# BEST AGILE ARTICLES 2018

Co-Editors: Michael de la Maza, CEC • Cherie Silas, CEC

# Best
# Agile Articles
# of 2018

**Editors:**
**Michael de la Maza, CEC & Cherie Silas, CEC**

## BEST AGILE ARTICLES OF 2018

Edited By: Michael de la Maza, CEC & Cherie Silas, CEC

Printed in the United States of America.

Cover design by Christopher Kyle Wilson
Book layout by Matt Kirilov
First Edition: September 2019

We are delighted to bring you this volume of the best agile articles of 2018. Our goal in publishing this book was to cull through the many articles that are published every year to bring you a curated set of high quality articles that capture the latest knowledge and experience of the agile community in one compact volume. Our purpose is twofold. First, we understand that it can be hard to figure out where to go when looking for ideas and answers. There are thousands of blogs, videos, books and other resources available at the click of a mouse. But that can be a lot to sort through. So, we thought we could be of some assistance. Second, we wanted to bring some visibility to many people who are doing good work in this field and are providing helpful resources. Our hope is that this publication will help them connect to you, the ones they are writing for. Our intention is that this publication is to be by the agile community as a service to the agile community and for the agile community. With that in mind, we pulled together a great group of volunteers to help get this work into your hands. The articles in this volume were selected by:

- A diverse Nominating Committee of nineteen people with expertise in a variety of areas related to agile.

- The agile community. A call for nominations went out in early 2019 and several dozen articles were nominated by the community.

The articles themselves cover a wide variety of topics including organizational structure, culture, and agile leadership. There is something for almost everyone here. This is the second book in the series. The first book, Best Agile Articles of 2017, is available on Amazon and as a free PDF.

We are thankful for the great participation by the agile community at large. If you would like to participate in delivering this publication in future years, contact us at the email addresses below.

Your co-editors,

*Michael de la Maza, CEC*

michael.delamaza@gmail.com

San Francisco, CA USA

*Cherie Silas, CEC*

cheriesilas@tandemcoachingacademy.com

Dallas, TX USA

# How Courage Can Create Safety

By Marsha Acker

Posted on February 28, 2018

Research shows that the number one contributor to team effectiveness is psychological safety. According to research by individuals like Amy Edmondson and the Project Aristotle study by Google, this means that it is critical to create a space where team members feel safe to take risks and be vulnerable in front of each other.

## Let All Voices Be Heard!

Leaders (that's you!) are able to use skills like facilitation and coaching to help create spaces where all voices can be heard and where people feel safe to take risks without fear of retribution. With leadership and guidance, it becomes the collective work of everyone on the team to create a safe space.

While psychological safety is something that we strive for in teams, it's not something that every team currently has. So I am often asked about what can be done in circumstances where safety is missing. My response is to encourage leaders to take the first step.

**What if our work as leaders is:**

- To be comfortable being uncomfortable?
- To take risks in service of others?
- To say what needs to be said, even if it feels scary?
- To find our authentic voice in order to help others see what we see?

## Name What You See Happening

One of the most powerful things you can do for a team is to name, in a morally neutral way, **what you see happening.** It might be to simply say, *"I'm confused about what direction we are going."* Other examples might include,

- *"I notice that we have been talking about this same topic for three weeks and that we have been unable to come to a decision."*
- *"I'm not sure what you want me to do; I need help."*
- *"I have things that I would like to contribute, but I wonder if they would be valuable here."*

## The Speech Act of Bystand

David Kantor calls this the speech act of “bystand.” It’s a vocal action taken in a conversation to bridge competing ideas or name what’s happening. It can be a powerful speech act for creating a shift in the conversation, but it is often underutilized or inactive in team communication.

Making a bystand is not about advocating for your solution, metaphorically poking someone in the eye, making a judgmental statement, personally attacking, or telling someone what’s “wrong” with their actions. It’s simply about naming what you see or what you are experiencing *in a manner that holds no judgment*.

When you model the speech act of observing without judgment as a leader, you help create a safe space for your team to join you in moving the conversation forward. Though it might feel uncomfortable at first, it is your demonstration of courage that can be an important first step in cultivating a team culture where diverse voices feel heard and acknowledged.

## Here Are Some Reflection Questions To Help You Take Action To Create Safety:

- **Have you had the impulse to say something 3 or more times?**I have a general rule about looking for patterns versus reacting in the moment. So notice what’s happening and look for a pattern.
- **What is your intention?** Does saying something further your own agenda or is it in service to the teams’ agenda? Both may be valuable, but be clear for yourself about which it is.
- **What’s at risk if you speak up?**Sometimes we create fear for ourselves by making up a worse outcome than what really might happen. Be honest with yourself about the answer to this.
- **What’s at risk if you remain silent?** This is about looking at the bigger picture. What opportunity might you or this team be missing?

## Your Turn!

Where do you need to be courageous today? Someone has to go first.

If not you, then who?

Marsha

***

To read this article online with embedded hypertext links, go here:

**https://teamcatapult.com/be-courageous-today/**

# About Marsha Acker

**Marsha** Acker is a leadership and team coach whose passion and expertise is helping leaders and teams identify and break through stuck patterns that get in the way of their desired performance. Marsha founded TeamCatapult, a coaching and change leadership firm, in 2005. She has over 20 years of experience designing and facilitating organizational change initiatives and believes that facilitation and coaching skills are core 21st century leadership skills. She served as the track chair for defining the ICAgile Coaching and Enterprise Coaching learning objectives.

Marsha is a Certified Professional Facilitator (CPF), Certified Professional Co-Active Coach (CPCC), Professional Certified Coach (PCC), Organizational and Relationship Systems Coach (Center for Right Relationship), and Dialogix - Certified Structural Dynamics Interventionist.

***

# Why Your Agile Teams are Bad at Estimation

By Heidi Araya

Posted on May 19, 2018

I get so many calls about estimation and forecasting. People reach out to me saying, "How can teams get better at estimating? Our forecasts are never accurate!", "The biggest pain point for us is predictability!" and "Are there better tools to show actual realistic forecasts on our Agile projects and initiatives?"

[To avoid the most obvious commentary about delivering value over focusing on output, let's assume that we have some validated feedback that this is solving a customer need.]

When I ask what the problem is, they begin to describe various scenarios...

- We have multiple teams working on an initiative, but only some of each team is working on it at any particular time
- One team has all the back end tasks and the other has all the front end tasks. And UX designers are not included in the estimates, as their work is a prerequisite, but does have regular rework required
- Some people working on initiatives are partially allocated
- Teams are interrupted with production issues or other responsibilities
- Teams gives estimates as if they were full-time dedicated, but they have many initiatives in progress
- Teams asked to produce estimates for new, higher priority items that supersede previously estimated initiatives (obviously, that is rendering the previous estimates & forecasts for work entirely obsolete, but conveniently, no one remembers these priority changes)
- Teams aren't stable; people come and go regularly to help on other things or contractors leave the program entirely
- Teams have dependencies in the sprint but have to pick up stories and tasks "to show progress" on something rather than sitting idle, or figuring out how to help unblock that work first
- Team is working on a brand new initiative and has never done this kind of work before
- The architectural vision or work to be done is unclear and teams are having trouble estimating with such lack of clarity
- Teams get feedback from the customer or product owner that changes the future work, but are still held to the original forecast by management
- Teams are striving to hit a pre-assigned date by management and are working as fast as they can, to the detriment of the code quality - or, the code is already nasty, and their estimates did not take into consideration the technical debt "mud" they have to wade through
- To add to the problem, in some cases managers don't want to reconfigure or create fully focused, autonomous or mission-based teams since it will disrupt the velocity on their existing team, in order to have the illusion of control or to chase the illusory forecasting predictability that everyone desires. (I ask, why is individual team velocity more important than delivering value to a customer?)

*Remember, an estimate is not a commitment,* *and a forecast is not a plan.*

Remember, an estimate is not a commitment, and a forecast is not a plan. And some people are very vocal that traditional estimation does not need to happen (see the #noestimates hashtag). I want to point out, that even if measuring throughput or story count for only those stories, the same problem will exist if you cannot know which, say, of the 100 stories will be worked on in a particular timeframe.

So if those problems exist, how can we set up the teams for success so that those estimates and forecasts, for better or for worse, actually mean something?

## Why Agile Works ... and How to Improve

However you decide to do it, and assuming you still feel the practice is valuable, estimation does assume that people understand the value of working in an Agile way:

- Small, measurable stories and tasks that are tested and complete and show actual forward progress (this greatly reduces risk of discovering things don't work later on)
- A cross-functional team that collaborates and actually owns an outcome versus people who work independently without regard to how that work interacts with others
- A team able to deliver something working at the end of a time frame or the ability to measure throughput and cycle time reliably
- A team able to raise impediments and get them resolved in a timely manner (an example of this might be where an environment is unavailable and the team cannot fix this issue themselves)
- The ability to inspect and adapt based on progress and feedback, which might change the forecast accordingly
- People dedicated full time to the team ... or at least work is proceeding without too many bottlenecks if they are not
- People are focused on minimizing the number of items in progress at any particular time
- Automated testing and unit tests, as well as environments similar to production and the ability to deploy regularly
- It also assumes that teams are motivated, proactive, want to do the right thing, and that they are trusted to do that work without being micromanaged.

So before anyone can even address any forecasting problem or perceived team problem related to story pointing, breaking stories down, or anything else, we must focus on the larger problem at hand... usually, it's an organizational lack of understanding about the principles that make Agile work.

*Usually, the problem is an organizational lack*
*of understanding about the principles*
*that make Agile work*

Obviously, no tool can "solve" this problem. It's the "way we work" that's the problem. Let's fix the problem, and forecasting will be much less painful.

From an allocation of humans perspective, Agile solves many problems that plan-driven work did not consider... that people are not machines, and that there is much lost in context-switching, over-allocating people, and assigning people to work on tasks without understanding the goal or vision for the whole. These things also had an impact on the ability to forecast and predict work. Agile solves these issues and creates environments where people are happy to come to work, in part because they can own an outcome, get stuff done, deliver things customers want, and contribute to the organization's mission.

See also: Happy People versus Getting Stuff Done?

## Public Service Advisory - a real statement on "What is Agile and What is Not"

"Forcing" teams to work in a specific way is, indeed, not "Agile." Let teams work out the way they should work themselves. The more prescriptive you are, the less you'll actually get out of them and the less they will be engaged.

Try giving them an outcome, and having them decide how to get there. There is a lot of guidance out there on how to do this.

- For background see: The State of Agile Software in 2018 by Martin Fowler
- See also: Agile Industrial Complex by Daniel Mezick

For tips on how to do start engaging people in agile without force, see

- https://ronjeffries.com/articles/018-01ff/imposition/
- www.OpenSpaceAgility.com
- https://www.agendashift.com

***

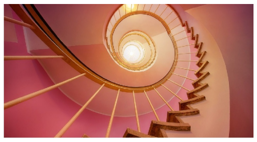
To read this article online with embedded hypertext links, go here:

**https://www.linkedin.com/pulse/why-your-agile-teams-bad-estimation-heidi-araya/**

# Why aren't my teams accountable?

By Heidi Araya

Posted on July 18, 2018



Managers, VPs, and execs ask me, "This thing in Agile where people are supposed to hold themselves and each other accountable is not working. My teams aren't delivering, and they don't care! How can I hold them accountable or get them to BE accountable?"

First: Though people often use the words interchangeably, accountability and responsibility are not the same thing at all.

*Accountability* refers to making, keeping, and managing agreements and expectations (taking account). And *Responsibility* is the feeling of ownership. According to Chris Avery, who has spent years thinking (and writing) about the differences between these two words, this is a sentence that might help clarify the point:

> *"If you have a manager and aren’t clear about what you are held accountable for, you might want to take responsibility for finding out."*

Most people really mean that they want teams and people to feel *responsible* for the quality of the work they produce. Let's revisit the idea of Agile teams, specifically Scrum teams, and being "responsible." Scrum teams should have everything they need inside their team in order to deliver something. According to the Scrum Guide:

> *"The Development Team consists of professionals who do the work of delivering a potentially releasable Increment of "Done" product at the end of each Sprint. A "Done" increment is required at the Sprint Review. Only members of the Development Team create the Increment."*

You can see where, if teams do not have things in their control or even a way to empower themselves, they might be reluctant to taking **responsibility** for that outcome and have a problem being held **accountable** for it also. If they don't have an environment to deploy to, for example, or the user stories do not capture some customer outcome--instead, there are handoffs across teams before the user story is complete--the idea of responsibility is lost and accountability can't even exist in this scenario.

Said another way: accountability is extrinsic; a feeling of responsibility is intrinsic.

## What does Done mean and what does it have to do with accountability?

> *When a Product Backlog item or an Increment is described as "Done", everyone must understand what "Done" means... members must have a shared understanding of what it means for work to be complete, to ensure transparency. This is the definition of "Done" for the Scrum Team and is used to assess when work is complete on the product Increment. - Scrum Guide, Nov. 2017*

If your user stories are not customer facing, and are just a bunch of tasks that people work on individually, ... "Well, I'm **done** with my piece of the work," they think. Now it's someone else's responsibility to wire up the logic, the UI, or deploy to an environment. The whole idea of "what is quality" is lost. We are no longer thinking of it in terms of the customer experience or problem we are trying to solve. In Scrum, we think of increments of tested, customer-facing functionality being completed inside a sprint. That was the whole problem with "plan-driven" or "waterfall" planning. The integration of much of the front and back end work was done late, testing was left to the end ... and no one took "responsibility" for what was working or not from a customer perspective. Figure out what **Done** really means for the business and make sure everyone knows.

## Commitment and Responsibility

Additionally, the Scrum Guide says the following:

> *"When the values of commitment, courage, focus, openness and respect are embodied and lived by the Scrum Team, the Scrum pillars of transparency, inspection, and adaptation come to life and build trust for everyone."*

How does this come into play in a business, especially the word "commitment"? Teams cannot feel committed to or responsible for delivering customer quality and value if they are only owning a piece of it.

Douglas McGregor, famous for his Theory X and Theory Y in the book, "The Human Side of the Business," posited that the managers are the ones who believe that Theory X people exist: these people are supposedly are just 'takers' who don't really want to work without being externally motivated by money or rewards. Because of this, Theory X managers believe all actions should be traceable to the individual responsible. This allows the individual to receive either a direct reward or a reprimand, depending on the outcome's positive or negative nature.

The problem with this, even aside from the terrible stereotype, is that work done in today's business relies on someone else to get things done; now more than ever. We don't just "read a ream of paper with requirements" and go off alone to deliver shiny, working software. That was rarely the case back in the day, and certainly not now. We rely on others everyday... our team members, the Product Owner, the people who set up and maintain the tools we work with, and even the sales people who sell things so we can continue working to deliver great products.

Let the teams feel committed to things inside their control. Empower them to feel commitment and responsibility by enabling them to do great work.

## How can I help my teams feel responsible?

- Has everyone read the Scrum Guide in detail, if teams are doing Scrum and understand the basics of "why Scrum works," and "how we do it here"? If not, read and understand it.
- Has anyone discussed with the team what "Done" means for them and for the business? If not, talk about it and agree on it.
- Has anyone set expectations of what the team should be able to accomplish, or not? Discuss it with the team.
- Does the team have clear goals and a purpose? If not, gather with the team and write them out on a page where everyone can refer to it.
- Do teams pair, mob, and share knowledge, or is it "every man for himself"? Change this by creating work that requires collaboration, you might be surprised at the change in behaviors.
- Have you set up the teams to be successful in their own right? Curious? Ask them if they have what they need to be successful, and what's stopping them.
- When they say they need something in order to do their best work, is this impediment removed (within reason of course)? Are themes across teams being raised up and organizational impediments removed? If not, look at what it will it will take to remove it, then start an organizational impediments backlog.
- Are teams holding valuable retrospectives, with actionable outcomes? If not, mix up the format or get a guest facilitator to run the retrospectives and see whether you can get one top thing per team to start working on.
- Are they allowed to cross train and learn if a skill set is missing in their team or are they told "no, that's someone else's job?" If no one has thought of this yet, ask the team what they need to be successful.
- Lastly, are you leaving time for your team to inspect and adapt ... and improve themselves and the company? If not, make dedicated time such as a half day every sprint, or a monthly improvement day focused on making it a better place to work overall.

There are so many opportunities for people to be happy at work and to feel responsible at the same time.

As for my feeling of responsibility? I felt responsible for sharing this information with you all today in order to make the world a better place for teams.

### Public Service Advisory - a real statement on "Why aren't my teams accountable?"

"Forcing" teams to work in a specific way is, indeed, not "Agile." Let teams work out the way they should work themselves. The more prescriptive you are, the less you'll actually get out of them and the less they will be engaged. Try giving them an outcome, and having them decide how to get there. There is a lot of guidance out there on how to do this.

- For background see: The State of Agile Software in 2018 by Martin Fowler
- See also: Agile Industrial Complex by Daniel Mezick

For tips on how to do start engaging people in agile without force, see

- https://ronjeffries.com/articles/018-01ff/imposition/
- www.OpenSpaceAgility.com
- https://www.agendashift.com/

***

To read this article online with embedded hypertext links, go here:

**https://www.linkedin.com/pulse/why-arent-my-teams-accountable-heidi-araya/**

# About Heidi Araya

**Heidi** partners with leaders and companies to help solve agile and digital transformation, teaming, organization design and strategy execution challenges. With over 20 years of experience in a wide range of industries and roles in technology, Heidi brings a pragmatic approach to her work. She collaborates with leaders and teams to create more responsive, effective, and resilient teams with engaged employees. Her passion is helping organizations harness the creativity and innovation of their people.

Heidi is currently Director of Agile Transformation at Tenable, a cybersecurity firm based in Maryland. She is also co-founder and advisory board member of the Open Leadership Network. She trains and speaks at events and conferences worldwide and co-hosts a popular virtual meetup series for Agilists at www.coachingagilejourneys.com. Connect with Heidi on Twitter https://twitter.com/HeidiAraya or LinkedIn https://www.linkedin.com/in/heidiaraya/.

# In Conversation with Edgar Schein: Answering Three Common Questions about Culture

By Aga Bajer

Posted on July 31, 2018

We all know that culture has the capacity to drive and alter everything that happens in our teams and organizations. Business results, services and products, relationships with clients and suppliers, the way people think, the stories they share, the way they go about their work and interact with one another; all this and much more is influenced by culture.

But culture can often be like a wet bar of soap—too slippery to grasp. To harness its force, we first need to understand its nature and dynamics.

And there's no better person to help us get a handle on organizational and team culture than Edgar Schein, one of the world's most well-known culture pioneers.

I had the pleasure to interview Ed shortly after his 90th birthday.

It was clear from the very start that in spite of officially becoming a nonagenarian, Ed has no intention of slowing down. He was just about to finish his new book, "Humble Leadership," and was audibly excited about the new venture he'd launched with his son, Peter Schein.

1

Our conversation was part of the CultureLab podcast series where I interview leadership thinkers, culture experts, entrepreneurs, and culture change agents to help listeners harness the force of culture.

Ed and I discussed many common concerns surrounding culture, and you can listen to the full interview here.

The CultureLab Podcast | Episode 14
The Culture Lab with AGA BAJER
When it comes to culture, you get what you settle for.
- Ed Schein -
www.agabajer.com

For this blog post, I've picked three common questions about culture that we addressed in the interview:

How is culture created?

Is it *really* possible to change culture?

Who is responsible for culture and for culture change?

## 1. How Is Culture Created?

The creation of a company is also the creation of its culture.

Culture comes to life at the very same moment the founder establishes business operations, irrespective of whether she is conscious or intentional about creating a culture.

It is the founder’s values, the norms by which she lives, the way she approaches her business, and the decisions she makes that shape the culture of her business.

If the founder is friendly and empathetic, chances are that the company culture will have a strong people focus. If the founder is competitive and controlling, the company culture will value status and hierarchy and prioritize winning. If the founder is highly analytical, the company culture will emphasize data-driven decisions.

People always observe the founder—how she thinks, what she believes, how she acts—and they infer culture from that.

The more successful a start-up is at the early stages of its existence, the stronger its culture tends to be.

The reason for this is very simple. As Ed Schein said during our interview, “Culture is what a group has learned in its history that has enabled it to survive and thrive (…) whatever values and norms enabled that group to survive and manage its internal affairs, they are its culture."

### In summary:

Companies are built in the image of their founders, and culture is always a reflection of how founders go about creating and managing their businesses. As a result, culture is the shadow of the founder.

## 2. Is it Really Possible to Change Culture?

I find that it often helps to think about culture as a living entity to answer this question. Just like other living entities, culture is hardwired for survival and self-preservation.

That’s one of the reasons why it can be so incredibly difficult to change an existing culture. Every attempt at culture modification is usually met with vehement resistance…coming from culture itself. After all, from culture’s perspective, it seems like a matter of life or death, and the term “culture change agent” sounds a lot like “culture assassin!”

In our interview, Ed talked about the fact that culture is a complex, multifaceted entity, consisting of many elements. It’s impossible and unnecessary to influence all of them at the same time.

Consequently, our goal should never be to change or overhaul the culture in its entirety. Instead, we should try to identify which aspects of culture might help or hinder the organization’s ability to survive and thrive—and focus on those.

And the more specific we can be about what issues, challenges, or problems we need to solve to help our business to thrive, the easier it will be to identify which aspects of culture will have to change.

Personally, I prefer to talk about culture evolution—not culture change. There are two reasons for that.

Firstly, “change” or “transformation” implies that there is an endgame we should be driving towards. However, as long as the internal and external environments keep changing, culture needs to keep changing, too. There’s no endgame in culture evolution

Secondly, we should be playing to culture’s strengths. And what culture is really good at is continuously evolving. Evolution is an integral part of culture’s survival mechanism

So here are a few steps towards a successful culture evolution:

Be very clear about what it is that you are trying to achieve in your business.

Identify key cultural enablers and key cultural obstacles towards the desired business outcome.

Make a plan to preserve, strengthen, and celebrate those aspects of culture that already enable your business goals.

Identify what needs to be in place to help evolve those aspects of culture that currently stand in the way of achieving your business goals.

Rinse and repeat as your culture and business evolve.

### In summary:

Culture will take care of itself whether we do something about it or not. It doesn’t want to change, but it will, and does evolve continuously to survive. Culture work can only be successful if it's linked to real business needs or a specific problem that needs to be fixed. This work can happen only when it focuses simultaneously on preserving the positive elements of culture and evolving those elements that need to change to help the business thrive. There's no endgame, only continuous evolution.

## 3. Who's Responsible for Culture and for Culture Change?

Evolving a culture in an intentional way might involve making changes in company structure, business model, and various supporting systems.

However, Ed stressed in our interview that nothing is more elementary and more powerful than what the leader does —or doesn't do—to embody the culture he or she wants to cultivate.

One of the biggest problems in organizations today is that while leaders state the need for change—for example, "We need to embrace a more participative leadership style" —they don't walk the talk themselves, or don't hold their people accountable for the desired behaviors.

***"You can only truly understand culture when you try to change it." - Ed Schein***

They try to "delegate" responsibility for culture to HR or even an external consultant and get frustrated when things don't go as planned.

But, there is no getting away from it— "It's the leader herself who should be doing the culture work, as she is the one who creates the culture through her own behaviors, what she rewards, and what she pays attention to," said Ed.

If you are a leader and what you want to see is more teamwork within your team or organization, you might have to ask your team members what they've done to improve collaboration on weekly basis. Perhaps consider introducing rewards for collaboration and setting team-based goals. If collaboration informs the way you organize your team, set targets, evaluate performance, and reward and promote your team members, you will inevitably see more collaboration.

The same goes for what you are prepared to tolerate—this will shape culture, too. As Ed said, "When it comes to culture, we get what we settle for."

## In summary:

Leaders are ultimately the ones responsible for shaping culture in their organization; culture is shaped by what they do, what they don't do, and what they choose to tolerate. For culture to evolve in the desired direction, leaders need to take action daily, talk about their expectations, walk the talk, and be willing to make difficult decisions when necessary.

My hope is that this blog makes a small contribution to demystifying culture, making it more accessible and less overwhelming in its complexity. It's only when we begin to understand culture that we stand a chance to harness its incredible power and start creating environments where people can do their best work.

To listen to my full interview with Ed Schein, click here.

*Notes:*

[1] Schein, E. (2018). *Humble Leadership: The Power of Relationships, Openness, and Trust (The Humble Leadership Series)*. Oakland: Berrett-Koehler Publishers.

***

To read this article online with embedded hypertext links, go here:

**https://www.agabajer.com/blog/106-in-conversation-with-edgar-schein-answering-three-common-questions-about-culture**

# About Aga Bajer

**Aga** Bajer, MD of Aga Bajer Culture Strategy Consulting

Aga works with companies around the globe, helping them map out a powerful culture evolution strategy that drives performance and business results.

She is the co-author of “Building and Sustaining a Coaching Culture”.

Aga is also the creator and host of one of the most popular podcasts on culture, The CultureLab with Aga Bajer.

The question at the core of her work is: “How can we harness the power of culture to create a world where people can do their best work?”

Aga blends experience as a leader, a consultant, a coach, a researcher, and an entrepreneur with her unstoppable and insatiable curiosity about what works and what doesn’t work in organisational and social change.

She worked with senior leadership of many major organisations, including the Hilton Hotels and Resorts, Citibank, Toyota, Porsche, GlaxoSmithKline, AstraZeneca, Leo Pharma, Wargaming, Sanofi, Amdocs, SAP, Heineken, the European Patent Office and others.

To join thousands of others in evolving their cultures, visit her website, www.agabajer.com, follow #HandleOnCulture or subscribe to Aga’s newsletter, The CultureLab Insider.

***

# Focus: Is it a Unicorn?

By Zsolt Berend and Tony Caink

Posted on December 11, 2018

Focus is hard enough as an individual, even harder in a team of brilliant minds. Why is it so hard?

. . .

Based on the study lead by Gloria Mark (at the University of California, Irvine) the bad news is that on average it takes **23 minutes and 15 seconds** to regain focus after switching to a new task. So now think about having not only one but multiple context switches in a work day, which is far from being unusual: the cost is just too high to ignore and yet we keep context switching, trying to multitask.

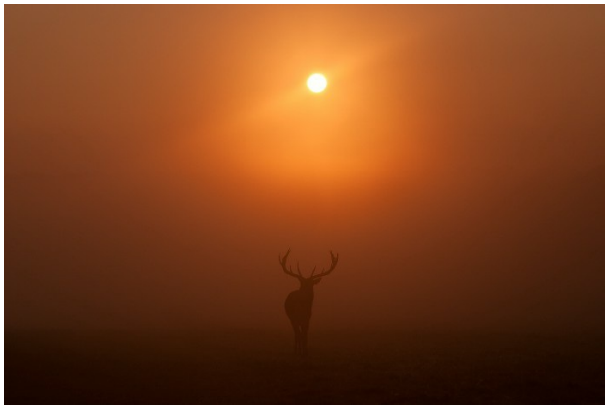
So why are we doing it?

Shockingly enough, neuroscientists suggest that our own brain actually works against us: it rewards the wrong behaviour.

As the studies point out, multitasking is supported by a dopamine-addiction feedback loop, effectively **rewarding** the brain **for losing focus**. Daniel J Levitin writes about how our brains have a **novelty bias**, meaning our attention can be easily distracted. This is also referred to as the “**bright shiny object syndrome**”

. . .

I was in Richmond Park taking this photo early morning just around sunrise. If you have never been, or not heard about this park, this is home of hundreds of magnificent free living deer. So I am there and witnessing a beautiful stag eating the grass and quickly abandoning it when our shiny, type G2 star rose above the horizon. The good news is that sapiens are not alone, stags are also suffering from bright shiny object syndrome :)



We also suffer of what’s called the “**busy trap**” of nowadays corporate culture. We **glorify “busyness”**: when you ask your colleague how are they doing, how is work, you most probably will get the answer of “busy”, “crazy busy” and the default response we give is “good to be busy”, “good problem to have”. This is a legacy of 19th century Taylorism when there was a linear, direct correlation between “busyness” and productivity which is unfortunately still applied in the knowledge workers’ world. Also, it is worth considering the correlation between wait time and utilisation which is best described in The Phoenix Project. When the utilization (busyness) goes past 80%, the wait time runs to the roof.

. . .

## Hacking the brain to reward the right behaviour

So how do we hack our brain to help ourselves and the team to turn our days into productive days as oppose to just busy days?

***1. Stop glorifying busyness, avoid the busy trap***

*Know your biorhythms*

Pay attention to your biorhythms (chronobiology). Keep balance between activity and being idle e.g. 90 min followed by 20 min. This is when you can have a coffee with Tony and think about whether focus is a unicorn or while you are idle dreaming discover the nuclear chain reaction like Leo Szilard did when the traffic light changed to green and he stepped off the curb at the British Museum on September 12, 1933 and the rest is history.

You can define time slots of no interrupt, deep work as part of the team charter, and agreed signals like headphones. Carl Newport lists many creative practices in his book, Deep Work.

*Introduce slack time*

Introducing slack not only prevents the problem of too much utilisation (above) but unleashes creativity and innovation. Tom DeMarco, in his book *Slack: Getting Past Burnout, Busywork, and the Myth of Total Efficiency*, makes the point that business kills innovation.

***2. Install transparency***

Visualize all things the team is working on, use simple boards.

*Walk the board*

Focus on the work not on the individual. Ask why an item is stuck on the board, who can you help, who can you pair up with, swarm to move the item to DONE. Alistair Cockburn defines product development, quite rightly as a “co-operative game.”

*Stop starting…*

To keep yourself and the team focused introduce limit the work in progress.

When pulling new work run through the decision tree first: “should I start this story”, “the story has value (a customer wants it)”, “the story is clear enough to work on (ready)”, “the system has (we have) capacity”. If any answers are No, then just say No. Stop starting.

*…and start finishing*

By finishing work you get two rewards: “getting things done” and “thank yous”. Communicating your results (whether they are positive or negative) means that others will recognise your work, resulting in more positive feedback. Praising and recognising the work of your colleagues can also increase your dopamine!

Dominica DeGrandis in her book Making Work Visible: Exposing Time Theft to Optimize Work & Flow talks about the problem with too much Work in Process (WIP), the negative impact on flow by not finishing or only partially finishing items and starting new ones.

***3. Set objectives at team level***

Establish team level objectives instead of individual ones. Reject work that does not contribute to team objectives. Seek the big reward that comes from sustained, focused effort instead of empty rewards. Focus on how great you will feel when your project is complete. A study of the University of Michigan professors found that results-driven focus motivated people to complete their work.

***4. Install Learning***

Articulate business objectives and outcomes with measures, build in learning capabilities. Leading indicators provide feedback, helps you tell whether you are delivering the right thing and helps you to sustain focus. Build measure learn.

• • •

Will you create a system to reward progress and real rewards or will you let bright shiny object syndrome destroy your productivity?

***

To read this article online with embedded hypertext links, go here:

**https://medium.com/sooner-safer-happier/focus-is-it-a-unicorn-5458801316a3**

## About Zsolt Berend

**Zsolt** is a business agility coach with 15 years hands-on experience in applying agile/lean ways of working principles and practices as practitioner, coach and trainer in a diverse portfolio of industry sectors, predominantly health care, telecom and financials services.

## About Tony Caink

**Tony** is a Transparency Installer helping teams to understand their work and optimise for flow and fun. He recently moved from a five year journey at one of the UK's leading Banks to an exciting mission to build something new and principled in a tier one consultancy. Just the latest step in a 20 year career focused on agile world domination. His family motto is 'keeping the 1970's alive' and he enjoys nothing more than a philosophical discussion with his 12 year old about the evils of taylorism.

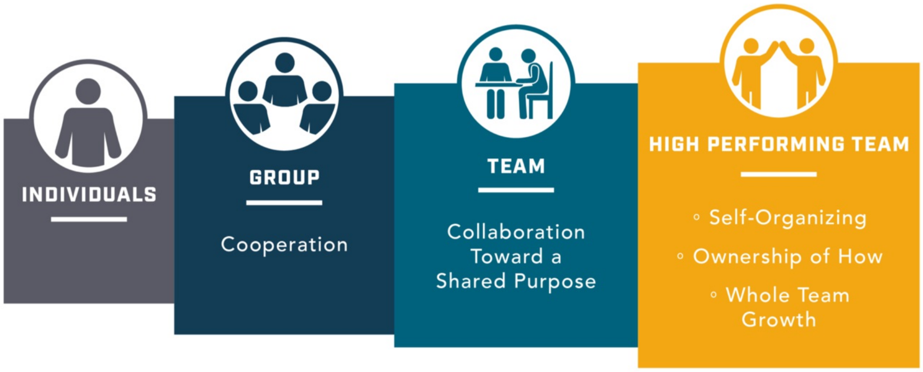
# Are leaders needed for self-organizing teams?

By Tricia Broderick

Posted on May 27, 2018

Last week, I explained my team evolution definitions including self-organizing teams. So today I'll focus on the questions similar to, "Are managers (executive, scrum master, functional manager, team lead, project manager, etc) needed for self-organizing teams?"

First let me highlight that I'm not getting into the discussion of manager vs. leader. A manager should be a leader. Bad managers do not represent manager roles. And labeling all managers as bad — well, I just expect more from people.



- **Leading Individuals**: When you have individuals, frequently leaders are spending quite a bit of time focused on: coordinating; building competencies (training); and building relationships/trust among people.
- **Leading Groups**: When you have a group, frequently leaders are spending quite a bit of team focused on: providing a shared purpose for cooperation; building competencies (training); building relationships/trust; and removing silos (increasing the need for collaboration by going beyond cooperation). For example, changing product backlog items to full value and having developers pair (front and back end developers).
- **Leading Teams:** When you have a team, frequently leaders are spending quite a bit of team focused on: celebrating collaboration; building confidence (mentoring); building relationships/trust; and removing single points of failure (including themselves) to begin to promote shared ownership. For example, cross-functional generalizing specialist training on an area that traditionally only one or two individuals owned.
- **Leading Self-Organizing Teams**: When you have a self-organizing team, frequently leaders have nothing to do. Kidding. That's ridiculous. However, the role has changed. Earlier, training and mentoring (with content knowledge) was common. Now, coaching and facilitating should be common. Earlier, you helped with competencies and confidence, now you support (sponsor/coach/facilitate) with continuing to elevate their knowledge, results, skills and teamwork. That may also still require training a new competency as they push themselves out of their comfort zone. The key is now leaders no sharing ownership by educating, engaging and encouraging their teams to invest in themselves, each other and the delivery of value.

Honestly, when I reflect back on helping various teams get to self-organizing: Individuals and Groups were the easiest (that is once I learned training and mentoring skills). Initially, teams were difficult because I had to personally learn new skills that would minimize and/or prevent me disempowering the team. Yet, self-organizing teams are by far the most rewarding and challenging for me as a leader. The results blew me away. The pride and team satisfaction were overwhelming. The appreciations up, down, and sideways were astounding.

So what was challenging? I had to inspect and adapt and face the unknown with them as the leader. I didn't know what challenge they were going to face next — how would I support them without taking ownership? Every self-organizing team I lead means uncharted new territory for me as a leader, which is both amazing and terrifying.

Do they need someone leading them every minute? Absolutely not.  In fact, if you are, you don't have a self-organizing team.  Would they be fine without a leader? Simply, this would never happen… because someone would fill that role whether it's assigned or not.

Is that a bad thing? Not necessarily, but it can lead to issues within the team that pushes them backwards in their journey if the person filling the role of leader doesn't have skills as a leader.  For example, a team that declares they don't need a ScrumMaster, yet someone on the team takes on those responsibilities — they have a leader whether they've assigned the role or not.

"But Tricia, sometimes multiple people do that." In my experience, it starts that way but quickly one person becomes the "not assigned" de facto lead.  If not having the title/role, helps, go for it.  But don't pretend that leading is not happening, or that the role should never be needed.

Personally, when this question comes up the most is based on this scenario:  The team wants to evolve to self-organizing, but the leader doesn't have the skills to help this happen.  The easy reactive pendulum swing answer is, "get rid of the manager."

Instead, I want to be a part of an organization that is investing in helping people learn the skills needed to lead beyond teams… because then our customers win, our organizations win, our teams win and individuals win.

That sounds like a much better long term solution to me.

***What have you experienced leading self-organizing teams?***

***

To read this article online with embedded hypertext links, go here:

**http://www.leadtotheedge.com/leaders-needed-self-organizing-teams/**

# About Tricia Broderick

**Tricia**, an Agile for All Principle, has more than twenty years of experience in software development and is passionately focused on facilitating high-performance environments. Her leadership at all levels of an organization helped lay the groundwork to shift teams from over one-year product cycles to feasible, daily delivery of quality value. Tricia openly shares stories from her firsthand experience to inspire people to reach new heights through continuous reflection and growth. As a principle in the Agile For All team and a member of the board of directors of Agile Alliance, she is dedicated to making a difference in the workplace. She is an exceptional leader, coach, mentor, facilitator, trainer, and a popular speaker at national conferences.

For more information, visit www.agileforall.com and www.leadtotheedge.com.

# How Your Organization Can Make Great Decisions By Default

By Tim Casasola

Posted on December 21, 2018

Photo credits to Rima Kruciene.

*First question.* What's the first thing you do when you wake up?

Do you check your phone? Read the news? Make a cup of coffee? Or hit the shower?

*Second question.* Do you have to tell yourself do that thing? Or do you do it without thinking?

My guess is that you do it without thinking because behaviors we do after waking up happen almost automatically. We check our phone, read the news, make coffee, or hit the shower without even thinking about it.

These automatic behaviors are our **defaults.**

## The power of defaults

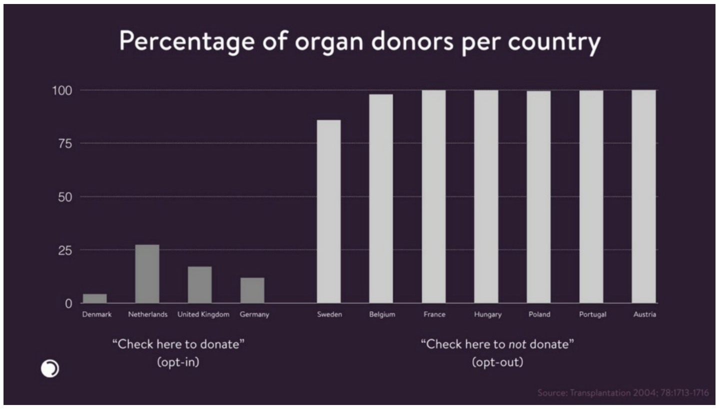
Most people think of defaults as the preselected settings in our software or devices. The default font on Microsoft Word is Calibri, size 11. Building thermostats are set to 72 degrees Fahrenheit by default. Default apps on the iPhone include Weather, Health, Calendar, Contacts, and Stocks. Unless we change the settings, the defaults prevail.

Here's the thing: People have defaults too. Our "defaults" are our preset behaviors and habits that happen almost automatically.

Our default behaviors are shaped by our digital and physical environment more than they are by our own willpower, determination, and good intentions. And if we don't make conscious choices about the behaviors and habits we want ourselves to do — *if we don't set our own defaults* — our environment will set them for us.

Take the 2003 study done by Eric J. Johnson and Daniel Goldstein. They compared countries that had opt-in organ donor policies with countries that had opt-out policies. Countries with opt-in policies asked "*Would you like to join the organ donor registry?*" while countries with opt-out policies asked "*Would you like to refuse to join the organ donor registry?*" The number of people who consented to being an organ donor was about **80 percent higher in**

**countries that opted-out** than those that opted-in.



Source: Transplantation 2004; 78:1713-1716

Another example comes from Vanguard's research. From looking at 401(k) participation rates, they found that 92 percent of employees participated in opt-out 401(k) enrollment, while 57 percent participated in opt-in enrollment.

Policymakers and economists recognize that defaults make it easy for people to make decisions they want to make but don't always follow through on. They also know that defaults make it hard for people to make unwanted decisions.

# Now, what if we applied the power of defaults to our organizations?

In large organizations, it might be common to…

- *Ask someone more senior for permission to make a decision.*
- *Keep computers open in every meeting to answer email.*
- *Create a thorough policy or approval process to prevent future errors.*
- *Make lengthy, comprehensive decks when pitching an idea to senior folks.*
- *Email people to find documents you're looking for.*

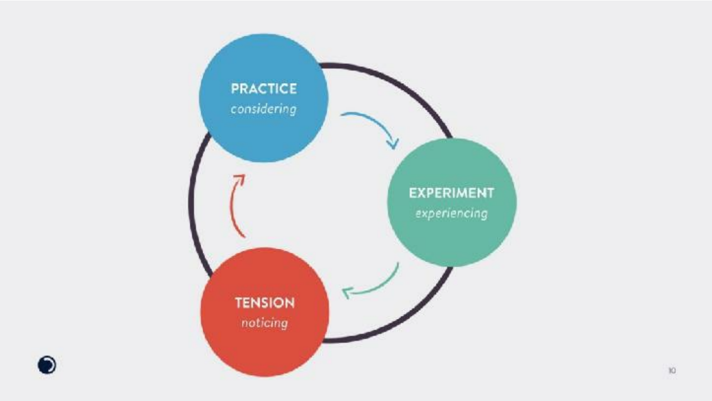
We don't challenge these practices, rules, and behaviors because it's status quo. "This is just the way it is here."

But what if your team was constantly challenging the organization's defaults that aren't serving them?

What if every team was open to the rest of the organization about defaults they are experimenting with?

What if your organization was intentional about the way they set up their environments so that they nudge people to make the good decisions they want to make?

Constantly experimenting with new defaults is a process. At The Ready, we think of this process as the Change Loop: a three-step continual pattern that enables continuous change.



The first step in the Change Loop is to notice **tensions**—obstacles in the way of your team/organization doing their best work. Then, consider a list of defaults (or **practices**) to that might address those tensions. From that list of defaults, choose one to two practices to **experiment** with. Then you're back to noticing tensions and the loop repeats itself.

This Change Loop challenges the typical big and flashy moves a typical organization does when trying to change. Moves like re-structuring the organization, announcing a completely new strategy, re-doing its entire brand identity, or hiring new executive leaders. However, only attempting big moves is the equivalent of signing up for a marathon when you've never ran a mile in your life. It's an effort that's doomed to fail.

The Change Loop also challenges the notion that organizational transformation has an end state. "*We just need to execute these four phases this year and voila, we're transformed!*" But take the best professional athletes. The best professional athletes don't think of themselves as "transformed" when they execute a certain amount of phases. They are constantly practicing, learning, and improving, as each improvement increases their motivation to improve even more. I've learned that the same goes for great organizations — they are constantly learning, improving their defaults, and evolving. *And this state of continual learning and evolving is transformation.*

# Some defaults we've seen work well with teams

Now that we understand the power of defaults, here are three defaults I've seen work well with teams we've worked with. Know that no default is perfect, and no default makes a team, individual, or organization perfect. Defaults simply offer us a way to continuously improve towards the standard that we aim for.

## 1) Flip your recurring staff meeting into an Action Meeting.

Most teams have a weekly or bi-weekly "staff meeting." Certain tendencies take place in this meeting: the manager shares updates to the team and ends up doing most of the participating, there's a pre-set agenda (usually set by the leader) for the meeting, and the conversation can go in any direction — brainstorming, problem-solving, or making a strategic decision.

I've learned that the weekly meeting works well if it has space for team members to get what they need to drive their work forward. This is why we advocate that teams who want to make their weekly meeting better try an Action Meeting. The Action Meeting is designed to create a shared understanding of the status of the team's projects, address obstacles and opportunities that are most important, and clarify the team's next steps (and the individuals owning each next step). You can read more about the Action Meeting in our in-depth post about it here.

The Action Meeting meeting has a learning curve, even for teams who are eager and fired up to change the way they meet. It's more process-oriented than your typical meeting. But from their first to third try, teams feel the positive difference an Action Meeting makes—from increased participation, to less back-and-forth tracking deliverables, to having less meetings because important issues are addressed, to even building better relationships.

### 2) Shift all of your internal communication from email to a transparent, asynchronous communication tool.

You’ve seen ways email can be a pain. Someone isn’t cc’d on an important conversation and now that person needs to be caught up. Backchannel conversations take place, which leads to a team not being on the same page. Updated versions of documents always get lost. And even though you spend most of your workday on email, the amount of emails in your inbox never seems to disappear.

At The Ready, we’ve seen teams experience a positive shift by moving their internal communication from email to a transparent, asynchronous communication tool like Slack or Microsoft Teams. This shift helps teams immediately experience what it’s like to *default to open:* to make communicating and working in the open the default behavior. No communication tool is perfect, but tools that are designed to promote transparency get teams to immediately experience what it’s like to work in an open way.

As an example, one experiment team at one of our clients embarked on a mission to make communication and information sharing more transparent in the organization. They were known as the info comms team. The info comms team along with two other experiment teams agreed to do a short experiment of using Slack instead of email for all communication related to the team’s work.

All three teams immediately saw the benefit: focused conversations, easier coordination, and transparent conversations. They felt the benefit of the shift so much so that the info comes team wanted to invite the entire organization to shift all internal communication from email to Slack. Wow.

So, the team proposed it to the organization. *We’d love to try a time-bound experiment: move all internal communication taking to Slack. By doing this, we hypothesize that information and communication will be more transparent in our organization.*

The shift to Slack lead to many important changes. A vibrant community and ongoing dialogue formed around the organization’s transformation effort. A #member-feedback channel was created, where anyone could share feedback they heard from their customers. Teams formed on their own to solve problems based on customer feedback. The senior leadership team even made their channel public so that anyone in the organization could read their conversations, ask questions, and chime in.

The organization knew they weren’t trying out a new tool for its own sake. They were clear that this provided a way to make working transparently the default.

### 3) Do a Closing Round at the end of every meeting.

Closing Rounds are one of the easiest ways to get into the habit of reflecting, learning, and improving. Here’s what you do: dedicate five to ten minutes at the end of every meeting to have everyone share their answer one-by-one to a specific prompt. Prompts like:

*“What did you notice about today’s meeting? What did you learn?”*

*“What went well? What could we do better?”*

*“What’s your biggest takeaway from today?”*

Most well-intentioned teams want to take more time to retrospect. But when the team gets busy, retrospection typically gets de-prioritized. This is why the Closing Round is so powerful: it’s an easy way to get teams to retrospect enough to the point where retrospection becomes a norm. And doing this in every meeting enables teams to continuously improve without even thinking about it.

# From here on out, you are a choice designer

Now that you know how powerful defaults are, you are officially a choice designer. As a choice designer, you must observe your team and organization's current defaults, challenge defaults that don't make sense, question where there's room for improvement, and place hurdles on unwanted behaviors. Most importantly, you must constantly search for ways to make it easy for you, your team, and your organization to make great decisions.

Our environment can make our choices for us. But we have the power to design it to make it easier for ourselves, our teams, and our organizations to make better decisions.

***

To read this article online with embedded hypertext links, go here:

**https://medium.com/the-ready/how-your-organization-can-make-great-decisions-by-default-950fa976e991**

# About Tim Casasola

**Tim** Casasola is a consultant who helps organizations, teams, and individuals succeed in the new paradigm of work. He’s a former Organizational Transformation Consultant at The Ready, where he helped companies such as Charles Schwab, GE, and Dropbox work in more adaptive and meaningful ways. He writes about the intersection of organizational and personal change on Medium, and publishes a weekly newsletter called The Jump. Tim is currently based in New York.

Website: timcasasola.com

***

# Ten Things the Beatles Taught Me About Being Agile

The Beatles included many secret agile tips in their songs.
St. Cohn's Mountain Goat Club Band
mountaingoatsoftware.com

By Mike Cohn

Posted on February 6, 2018

While cleaning my home office recently, I decided I would listen to all of the Beatles' albums. I'd start with "Please, Please Me" from 1963 and work my way to 1970's "Let It Be." My goal was to finish cleaning before the Beatles ended with "Get Back." In doing so, I realized that much of what I know about agile, I learned from the Beatles. In particular, here are the top ten things I learned from Beatles songs.

## 10. Eight Days a Week

In this song, John Lennon sings to a girl that he "ain't got nothing but love, Babe, eight days a week." Of course, there aren't eight days in a week, so John is really singing about overtime. And from it, I learned that while overtime is best avoided in most cases, it may occasionally be needed when a regular work week is not enough to show I care.

## 9. Don't Let Me Down

In this song, John assumes the persona of a user or customer of an agile product. And he reminds us that our goal is “don’t let me down.” To ensure that we delight our customers, our agile team must remain focused on delivering value.

## 8. Come Together

One of the best ways for a team to deliver value to its customers and users is to collaborate with them. This led the Beatles to encourage teams and their stakeholders to “come together” to build the best product possible. In particular, Lennon condemned developers who ignore user's needs and just build what they want. He called such developers "jokers" and sang, "Got to be a joker, he just do what he please." The best agile teams do not have jokers who just build what they please.

## 7. Tomorrow Never Knows

The popular agile phrase, "You ain't gonna need it" is often simplified to YAGNI. This refers to the difficulty of designing for more than the immediate future. Before we had YAGNI, we had John Lennon warning us that "tomorrow never knows," meaning we should design only for the current day.

## 6. We Can Work it Out

This collaboration between Lennon and Paul McCartney suggests to teams and stakeholders that when it comes to solving problems, “we can work it out” when we work together.

The line, "Do I have to keep on talking till I can't go on?" reminds teams that code settles arguments. Build it, release it and see what customers think rather than talking until you can't go on.

By cautioning teams that "only time will tell if I am right or I am wrong," Lennon and McCartney urge teams to create a minimum viable product to find out how reality matches up against expectations.

## 5. I Am the Walrus

In this song, Lennon was singing about what is commonly known as eating your own dog food. The idea here is that team members should, if at all possible, use the product they are building. This means that developers are not just developers but are also users. Or, as Lennon sang, "I am he as you are he as you are me and we are all together."

## 4. A Day in the Life

In this poignant tune, the Beatles caution that only by watching a user as he “woke up, fell out of bed, and dragged a comb across his head” could agile team members truly understand their users. To develop products that fully meet their users’ needs, agile teams need to get out of their offices and to go study a day in the life of their users.

## 3. With a Little Help from My Friends

The daily standup is one of the most common practices among agile teams. During these meetings, team members are encouraged to present any impediments they are facing and to solicit help from others on the team. Those working on an agile team quickly learn they can “get by with a little help from [their] friends.”

## 2. Can't Buy Me Love

In the Beatles' fourth number-one hit, Paul McCartney reminded agile teams that there are some things money can't buy. Many fans heard McCartney sing, "I don't care too much for money, money can't buy me love," and thought he was referring to the love of a woman.

Agile teams understood him, however, to be singing about the love of customers for the team's product. McCartney was admonishing teams to produce high quality software. He knew that even one low quality release would cause customers to no longer love a product and that no amount of money could be thrown at poor quality to buy the customers' love.

## 1. Getting Better

Even the best of agile teams can get better. The pursuit of continuous improvement is common to all of the best agile teams. McCartney knew this and wrote this song to encourage teams to always be “getting better, a little better all the time.”

There you have it--the top ten things I've learned from Beatles songs.

No wonder they were known as the Fab Four. Beyond the great music and great lyrics, I bet they could have developed some amazing products had they set their minds to it.

## What Do You Think?

What do you think? Are there Beatles songs I missed? Are there songs by any other artists that hold agile lessons? Please share your thoughts in the comments section below.

***

To read this article online with embedded hypertext links, go here:

**https://www.mountaingoatsoftware.com/blog/ten-things-the-beatles-taught-me-about-being-agile**

# Ten sentences with all the Scrum Master advice you’ll ever need

## 10 Practices to Take to Heart

By Mike Cohn

Posted on May 22, 2018

TOP 10
Want to be a great Scrum Master? These are the ten best pieces of advice I’ve ever given or received.
mountaingoatsoftware.com

Do you want to be a great Scrum Master?

I hope so. (Well, unless you’re a product owner or in some other role!) I’ve spent over 20 years as a Scrum Master. Over that time, I’ve given and collected quite a lot of advice. I’ve distilled it down to the ten best bits for you.

### 1. Never Commit the Team to Anything Without Consulting Them First

As the Scrum Master, you do not have the authority to accept change requests (no matter how small) on behalf of the team. Even if you are absolutely positive that the team can fulfill a request, say, “I need to run this by the team before we can say yes.”

And certainly don’t commit the team to deadlines, deliverables, or anything else without first talking to team members. You may not need to talk to the whole team--plenty of teams will allow some or all members to say, “Yeah, we can do that” without a whole-team meeting. But it’s still their decision, not yours.

### 2. Remember You’re There to Help The Team Look Good

Being a Scrum Master is not about making yourself look good. You look good when the team looks good. And they look good when they do great work.

You know you’re doing your job well when those outside the team start to wonder if you were even needed. Yes, it can be scary if your boss wonders if you’re necessary. But a good boss will know that your skill and expertise make you appear unnecessary when in fact you are indispensable.

Trust your manager to understand the difference between looking unneeded and being unneeded.

### 3. Don't Beat the Team over the Head with an Agile Rule Book

Neither Scrum nor agile comes with a rule book (though some have attempted to create one).

If your product has users, consider writing user stories. But stories aren’t required to be agile. If someone needs to know when you’ll deliver: estimate. If not, maybe you don’t. If you think an end-of-sprint review is too late to receive feedback, do one-at-a-time reviews as each feature is built.

Being agile is about honoring the principles and values that create agility. If you stay true to those, you can't go too far astray, regardless of what some may tell you.

## 4. Nothing Is Permanent So Experiment with Your Process

Part of honoring the principles of agility is to experiment with your process. Encourage the team to try new things.

Does your team love two-week sprints and think they're working perfectly? Great. Now ask them to try a one-week or a three-week sprint and observe the results. Experiments might not always be popular, but they are the best way to ensure that you continue to uncover new, better ways of working.

## 5. Ensure Team Members and Stakeholders View Each Other as Peers

Team members and business-side stakeholders each bring an important perspective to a product development initiative. As such, each needs to be valued equally.

When either side views the other as something to be tolerated, the organization as a whole suffers. Development teams need to understand the unique perspective brought by stakeholders. And stakeholders need to respect the development team, including listening when developers say that a deadline is impossible.

## 6. Protect the Team, Including in More Ways than You May Think

Perhaps the most often given agile advice is that a Scrum Master needs to protect the team from an overly demanding product owner or stakeholders. And that's good advice. Sometimes product owners simply ask for too much too often and too aggressively. This forces teams into cutting corners, usually quality corners, that come back to haunt the project.

And so a good Scrum Master protects the team against this.

But what you don't hear as often is that a good Scrum Master should also protect the team against complacency. Good agile teams seek constantly to improve. Other teams settle, perhaps unconsciously, into thinking they've improved enough. And they likely are dramatically faster and better than before they'd heard of agile. But even great teams can often become even so much better.

Great Scrum Masters protect teams from ever feeling they've got nothing left to learn.

## 7. Banish Failure from Your Vocabulary

Every now and then I'll visit a team that refers to a sprint as a "failed sprint." Usually this means the team didn't deliver everything they planned. I hardly consider that a failure, especially if the team finished most planned items or if they deftly handled an emergency.

When a basketball player shoots the ball toward the basket and scores, it's called a field goal. When the player misses, it's called a field goal attempt. Not a failure. An attempt.

Good Scrum Masters help teams adjust their thinking so that they recognize sprints and features that fall short of expectations as attempts rather than failures.

## 8. Praise Often But Always Sincerely

The other day I told my teenage daughter that I was proud of her. Her face lit up. That shouldn't have surprised me. Who wouldn't like to be told someone is proud of them?

But the way she reacted made me realize I must not tell her this often enough. I thought it was equivalent to me telling her something obvious, such as, "You're tall." But I learned it wasn't.

Don’t ever offer false praise. No one wants to hear that. But when your team members do good work, let them know. Chances are, they aren’t hearing it often enough.

## 9. Encourage the Team to Take Over Your Job

A team that is new to agile will rely on their Scrum Master or coach in significant ways. The team may not know how to keep daily scrum meetings under fifteen minutes. Or they may not understand the importance of overlapping work or of being a cross-functional team.

The same is true of a an inexperienced sports team. The coach of the little kids learning to play football (soccer) needs to teach them everything. When my daughters were 6, their coach would run along the sideline the entire game yelling, “Kick and run!” If he didn’t, the young players would forget. Even with him yelling, occasionally some kid would just sit down on the grass and stare.

Contrast the coach of the young kids with the coach of a World Cup team. On a World Cup team, players have learned what to do. If the coach is late for practice, the players will know what drills or exercises to start the day with. The World Cup coach doesn’t need to remind the players to kick and run. But the World Cup team would never tell you they don’t need a coach at all.

No matter how good an agile team gets, I still think they benefit from having a Scrum Master or coach. But good agile teams take on some of the more straightforward coaching tasks themselves as part of their own journeys to mastering the skills needed in product development.

## 10. Shut Up and Listen

Some of the best coaching or mentoring you’ll do is to stay silent and let the team figure out the answer.

This can be hard. When you see your team struggling to figure out what to do, it’s natural to want to jump in and offer advice. But if you solve problems or even offer suggestions too readily, team members learn to just wait for you to solve every problem for them.

I don’t want to imply you can’t ever offer suggestions. You’re a smart person. If not, you wouldn’t be in the role you’re in. But part of being a great Scrum Master is helping teams learn how to solve problems on their own. If you solve every problem team members face, they don’t get a chance to learn how themselves.

## What’s Missing from this List?

I’m sure I’m missing some pearls of wisdom. What is some of the best one-sentence advice you’ve received or given as a Scrum Master? Please share your thoughts in the comments below.

***

To read this article online with embedded hypertext links, go here:

**https://www.mountaingoatsoftware.com/blog/ten-sentences-with-all-the-scrum-master-advice-youll-ever-need#.W3dVCm4tRtQ.linkedin**

# Why Agile Teams Should Estimate at Two Different Levels

It's useful to estimate both the product and sprint backlogs. But for different reasons and in different units.
mountaingoatsoftware.com

By Mike Cohn

Posted on September 11, 2018

It is very common for agile teams, especially Scrum teams, to estimate both their product backlog and sprint backlogs. In this article, I'll address:

- Why estimating both the product backlog and sprint backlog can be useful even though it seems redundant
- Why teams should estimate the two backlogs in different units
- When teams should estimate
- Whether all teams should estimate

If you're new to agile or need a quick reminder about what the product and sprint backlogs are, you can watch these two videos on the product backlog and sprint backlog.

## Why You Should Estimate both the Product Backlog and Sprint Backlog

It's useful to estimate both the product backlog and the sprint backlog because the estimates are used for different reasons.

### Reasons to Estimate the Product Backlog

There are three main reasons to estimate a product backlog. First, it allows a team and its product owner to make longer term predictions about how much can be delivered by when. It allows teams to answers questions like:

- How much can you deliver in three months?
- When can a certain set of product backlog items be delivered?

Second, it aides product owners in making prioritization decisions. Priorities should be set based on the expected benefits and costs of the product backlog items. A product backlog item estimated at 3 story points, days, or whatever units you use, is typically a higher priority than it would be if it were estimated at 100.

To say this isn't the case would mean that you always order the best wine on the wine list or drive the best car manufactured regardless of price. Chateau Mouton-Rothschild 1945, anyone?

Most of us, including product owners on agile projects, cannot afford to make decisions that way. In prioritizing work, we consider the cost of developing product backlog items. For most items, the estimate of the effort involved is the biggest component of the cost.

A third reason to estimate items on the product backlog is that team members become more knowledgeable about the item by thinking about it enough to estimate it. This means there will be fewer surprises when the feature is being developed.

### Reasons to Estimate the Sprint Backlog

Now let’s look at why teams should also estimate the sprint backlog. The are two reasons to estimate the sprint backlog. First is that it helps the team determine how much work to bring into the sprint.

By splitting product backlog items into small, discrete tasks and then roughly estimating them during sprint planning, the team is better able to assess the workload. This increases the likelihood of the team finishing all they say they will.

Second, identifying tasks and estimating them during sprint planning helps team members better coordinate their work. For example, if sprint backlog items are not estimated, a team might not notice a critical path through the work or that the designer will be the busiest during the coming iteration.

## Why Estimate In Different Units

Because the estimates on a Scrum teams two different backlogs serve different purposes, they should be made in different units.

For product backlog items, in particular, it is vital that the team can estimate quickly.

To see why, suppose a boss asks the team to estimate when forty product backlog items in the form of user stories can be delivered.

This could be an entirely valid request. Perhaps the boss wants to know whether to hire additional team members if the project will take too long. Or perhaps the boss only wants to start the project if it can be reasonably expected by a specific date.

If the team were to answer the boss by splitting each user story into tasks, as is commonly done in sprint planning, and estimating each of those, the time spent estimating would be huge.

If we assume an average of 15 minutes discussion and estimating per user story (as is commonly needed during sprint planning), estimating 40 user stories would take 600 minutes or 10 hours of whole-team effort.

If the team can instead use higher-level but equally accurate estimates on the product backlog items themselves, those estimates can usually be created much more quickly. I advise teams to target three to four minutes on average per product backlog item. In this case, estimating 40 user stories would take no more than 160 minutes, or about 2-½ hours.

The best way to do this is for a team to estimate its product backlog items in story points and its sprint backlog tasks in hours.

This works well because story points are a more abstract measure that individuals with different strengths can agree on. Just as you and I can agree on the length of the measure “one foot,” even though our individual feet are most likely different lengths, so can agile team members of different skills agree that this user story will take twice as long to do as that user story.

Story points don’t work, though, at the sprint level. During sprint planning, remember the goal is for a team to determine how much work to bring into the sprint. That’s hard with abstract units like story points. It’s far easier with hours.

Estimating in hours is feasible on a sprint backlog because sprints typically contain fewer items than the entire product backlog, which means it won’t take as long. Plus, a typical sprint task will be performed by one person. And in many cases, it’s clear which person will do it. These factors make it feasible to estimate a sprint backlog in hours.

## When to Estimate the Product Backlog

It’s clear that sprint backlog items should be estimated as part of a sprint planning meeting when the sprint backlog is created. But when should a team estimate its product backlog items?

I recommend estimating product backlog items at two different times. First, estimate a day or two after holding a story-writing workshop.

This is a meeting I recommend product owners conduct for their teams on a (roughly) quarterly basis. The goal is to identify the user stories needed to achieve some larger-than-a-sprint initiative. Identifying those product backlog items could take 2-4 hours (per quarter). Estimating them should then take an hour or two more.

The second time a team should estimate product backlog items is once per sprint, if new product backlog items have been added since the previous sprint. This can happen any time but it should be relatively late in the sprint to minimize the chance of new stories coming in afterwards. Most commonly this is done during a team’s product backlog refinement meeting or immediately following a daily scrum when everyone is already interrupted and present.

## Why Not Estimate Product Backlog Items During Sprint Planning?

It may seem like a good idea to estimate product backlog items right at the start of the sprint planning meeting. However, there are two big problems with this.

First, it’s too late for the product owner to consider the estimate when prioritizing.

Remember that one of the reasons why teams estimate their product backlog items at all is so that the product owner can prioritize. If the product owner isn’t given estimates until the start of sprint planning, it’s not realistic to assume the product owner will fully consider those when prioritizing.

Second, teams that estimate product backlog items at the start of their sprint planning meetings tend to spend much longer estimating.

I suspect this is because team members are about to perform more detailed sprint planning. With their minds on that, the need for more detail often creeps into the effort to estimate the product backlog, making it take longer than my target of 3-4 minutes per item.

For these reasons, try to estimate any new product backlog items that need to be estimated outside of sprint planning and also late enough in the sprint that most (if not all) new user stories have already been identified.

## Should All Teams Estimate

I’ve established that there are good reasons to estimate both the product backlog and the sprint backlog. And I’ve argued that these estimates should be in different units (story points and hours) and should be estimated at different times.

But do these arguments apply to every agile team? Or are there some teams who don’t need to estimate either or both backlogs?

I’ll share my thoughts on that in my weekly tip on Thursday. If you haven’t already, you can **sign up to receive a**

**short tip** on succeeding with agile from me each Thursday.

## What’s Your Experience?

How does your team estimate its product and sprint backlogs? Do you use the same units? When do you estimate? Please share your experience in the comments below.

***

To read this article online with embedded hypertext links, go here:

**https://www.mountaingoatsoftware.com/blog/why-agile-teams-should-estimate-at-two-different-levels**

# About Mike Cohn

**Mike** Cohn specializes in helping companies adopt and improve their use of agile processes and techniques to build extremely high-performance teams. He is the author of User Stories Applied for Agile Software Development, Agile Estimating and Planning, and Succeeding with Agile as well as the Better User Stories video course. Mike is a founding member of the Agile Alliance and Scrum Alliance and can be reached at hello@mountaingoatsoftware.com. If you want to succeed with agile, you can also have Mike email you a short tip each week.

# The 3 Principles Agile Leaders Should Live By During An Organizational Transformation

By Eric Cottrell

Posted on November 9, 2018

Today's Agile leaders manage with a more complete arsenal of skills and techniques than leaders of the past who led fundamentally different teams. For Agile organizations, the days of top-down hierarchy and autonomous decision making have joined the floppy disk as obsolete and outdated concepts in modern business. Leaders still can be effective without following Agile business development principles, but those who embrace the Agile methodology achieve greater success, improve the customer experience, and boost employee engagement.

Modern Agile leaders now see themselves as catalysts supporting their teams' and organizations' efforts to accelerate towards organizational agility and increased efficiency. While leaders still exhibit solid situational and personal leadership skills that make them strong individually, they're adopting key new characteristics and exhibiting forward-thinking traits that set them apart.

## What Modern Agile Leaders Get Right

Agile business leaders follow these key principles.

1. **Optimizing the whole.**

   All too often, leaders understandably don't fully appreciate the significant change they personally can affect across an entire organization. They can have an outsized positive impact. By necessity, they tend to think within the existing framework of the people who work directly for them and report up to them–and there are often enough troubles there. However, this view limits the big impact they can make across their organization.

   Modern leaders strive to create value across the entire company rather than just the department or groups under their jurisdiction. Doing so requires working across different departments and working closely with other leaders and their teams to optimize work. The leaders who are able to gain enough altitude to see (or imagine) work flowing across organizations, from hand-off to hand-off, until that work reaches the customer, will likely find significant opportunities for improvement that ensure the whole company is delivering increased value to its customers.

   Optimizing the whole allows the groups that are best able to solve problems to work horizontally across different departments to deliver the best products or customer experiences. This requires leaders to build durable relationships with key colleagues, personnel, and other managers to ensure they craft positive experiences, jointly work out challenges, share disappointments, and have the necessary hard conversations about how the company can improve as a whole.

2. **Personal Leadership and Context Awareness**

   Agile leaders are embracing a new level of introspection about themselves, their leadership approach, and their role within their company. They do so willingly because they've bought into a radical idea that by empowering their people they are harnessing far more capability, wisdom, and horsepower than they could ever hope to have by themselves. This is a massive shift from a "hero leader" who does it all, to the catalyst leader who empowers her teams. This change can begin with a simple personal leadership assessment that will remind leaders of what they value, what matters, and what their role is.

   Personal leadership questions include:

   - Why am I here?
   - Why am I investing so much in the place at this time?
   - What do I hope to accomplish?
   - What am I going to do about it?

   The answers to these questions serve to remind the Agile leader of what motivates them and what they care about. That has significant benefits to everyone around them, but mostly, to the leader herself.

   Now the truth is leaders–Agile or otherwise–often find themselves in very tough spots. Clients are upset, leadership is growing impatient, work isn't being delivered on time, etc. There's no end to the challenges. So how does a modern leader respond? Again, rather than simply passing along the angst, anger, or abuse, modern leaders are learning to gain context, to find root causes, and to make sense of the cacophony of noise swirling around them.

   Gaining context awareness can be a huge asset, even in the midst of a crisis. Again, there are a few questions that leaders can ask to gain valuable perspective:

   - What is really going on, and what can I/we do about it?
   - Do we collectively have the skills and competencies needed?
   - How will my and my team's actions affect those around me?

   By dedicating time to answering these questions, the leader can see how to tackle the problem utilizing the team's strengths and mitigating weaknesses. Rather than rushing in, can offer a situational view that will provide the whole team breathing room to think more clearly about how to take effective action swiftly.

3. **Effectively Empowering Talent by Managing With Intention**

   Author David Marquet, former commander of the USS Santa Fe nuclear submarine, discusses the impact intentional leaders can have on every level of an organization in his book, "Turn the Ship Around."(It's one of my favorite leadership books!)

   Marquet discusses how intentional leaders empower employees to make smart, sound decisions at the origin of insight and seek the best understanding of problems and opportunities. Managing with intention provides a frame of reference to create intentional employees who are equipped to be leaders themselves who in turn manage with intention.

   While I suggest you read Marquet's excellent book, here are three of the main skills necessary to lead with intention:

   - **Clarification:** Make sure all team members fully understand the scope of their roles.
   - **Competency:** Ensure team members have the ability to do their work, solve problems, and make sound decisions.
   - **Certification:** Have employees closest to problems discuss the different angles and dimensions of the problem that are critical for success.

Leading with intention shifts management from a top-down decision-making framework to a bottom-up hierarchy. Employees at the root of problems grapple with issues and propose various solutions. They are fully empowered to overcome challenges and only require a leader to approve their decisions and provide additional insight or guidance. The entire team leads with intention, which saves valuable time since leaders aren't working through problems alone and can rely on their teams to overcome challenges. Marquet calls this "leader-leader," and it's powerful.

Leaders often think their primary duty in the game is to block and tackle and make hard decisions. Intentional leaders, however, share certain decision-making rights by allowing team members to evaluate, judge, and assess problems with all the right thought behind it. Intentional leadership allows individuals to take ownership of challenges rather than roll problems upstairs to senior executives. This shift empowers individuals at the point of problems to solve crucial issues, eliminates cumbersome chain-of-command hierarchy, increases speed and efficiency, and makes your company and teams more agile.

Intentional Agile leaders share many traits with their peers, including confidence, integrity, empathy, honesty, and accountability. They also dedicate time to thinking about how they can best prepare the people and teams closest to problems to be ready, willing, and able to solve challenges and to properly inform supervisors of their results.

To learn more about Agile leadership tips and tools, explore our Transformation Library.

***

To read this article online with embedded hypertext links, go here:

**https://agilevelocity.com/leadership/the-3-principles-agile-leaders-should-live-by-during-an-organizational-transformation/**

# About Erik Cottrell

**Erik Cottrell** is the Senior Vice President of Client Success & Marketing at Agile Velocity. With over 20 years of experience in Product Management and leading strategic growth, Erik now leads his team in developing new products and initiatives with the goal of delighting customers.

Erik believes the path to lasting high performance goes beyond process change and champions that a uniquely agile mindset informs culture, technology experiences, cross-department collaboration, and employee engagement. Applied holistically and pragmatically, agility becomes a key competitive advantage.

# Virtual Reality Will Disrupt Agile Coaching and Training

By Michael de la Maza and Elena Vassilieva

Posted on March 21, 2018

### Key Takeaways

- Online technology will disrupt agile coaching and training in the next 3-5 years.
- The agile/Scrum community has been slow to embrace online technology because the Manifesto emphasizes face-to-face interaction.
- Online technology promises to improve learning outcomes and lower costs for an increasing number of use cases.
- Early successes include an online agile user group and the use of a 360 degree camera to support team-level training and coaching.
- By the end of 2020, at least one large, credible agile/Scrum certification organization will be running agile/Scrum certification courses in virtual reality.

We believe that online technology (virtual reality, augmented reality, adaptive personalized learning and videoconferencing) will disrupt the agile coaching and training spaces in the next 3-5 years. This will have a significant and palpable effect on the 1M+ agile team members, 10K+ agile coaches, and 1K+ companies which are currently involved in agile efforts. We believe that business models will change drastically, winners will become losers (and vice versa), and the skills needed to lead agile transformations will change significantly.

In this article, we focus on virtual reality and describe its history and current capabilities. Our work is informed by a case study at a 500 person, billion dollar public company in which nascent virtual reality capabilities caused C-level executives to better understand the impact that their work was having on the organization. We end with predictions about what we believe will happen in the next 3-5 years.

### Where are we now?

Online technology has a significant and growing effect on the broad world of education and training. There are now residential college programs in which some of the courses are taught online, a bachelor’s degree from a top university can now be obtained entirely online, and all Ivy League universities offer online courses. In addition, there are many online professional programs, particularly for people in the field of software development.

One of us (Michael) spent five months coaching at edX, an edtech nonprofit started by MIT and Harvard. edX has three goals, one of which is to learn about learning. With tens of thousands of students taking its most popular classes, edX has more information about student behavior than even the most popular university professors. This has created an environment in which edX courses have rapidly improved in a way that live courses taught by professors have not.

We believe that something similar will happen with technology mediated agile training and coaching. Because technology allows for extreme instrumentation of activities, behaviors, and results, the developers of online coaching and training will learn what works and what does not at a much faster rate than their face-to-face counterparts.

Today, there are already clear use cases (e.g., facilitating distributed teams) in which online technology plays a clear role. These use cases will serve as petri dishes in which online technologies will be generated, tested, evaluated, and selected. The best ones will then go on to challenge face-to-face agile coaching and training.

One such use case are online special interest groups. In 2017, Alexander Frumkin started the Agile Practitioners Online Special Interest Group. This group has a format very similar to an in person agile user group but it meets entirely online. It has grown from approximately ten registrations per session to 50 registrations per session in a little over six months. We expect that online agile user groups will form to support underserved geographic areas and subjects.

Adaptive personalized learning technologies which modify the content of a course based on what the learner needs, have been shown to support learning outcomes that far exceed those of a standard online course. Yigal Rosen of Harvard University and Rob Rubin of Microsoft have open sourced an adaptive learning framework and it is only a matter of time before it is applied to online agile training. Rob Rubin says "Project ALOSI points to a time in the near future where learners will be served content that is appropriate for where an individual falls on his or her personalized learning curve, as well as coaching that reinforces their growth mindset. Cognitive and content based delivery is key to incremental learning and hard-to-measure skill acquisition."

Virtual and augmented reality is the next step in the evolution of online educational technology. These environments enable realistic simulations and a "felt" experience that drives learning. Agile training and coaching with its emphasis on games and visualization readily lends itself to virtual reality.

Today's virtual reality products are not accessible to all consumers. Virtual reality headsets cost approximately $500 and they require high end computers. However, we expect the price to rapidly come down and, in the same way that all phones now have high resolution video, we expect that all consumer laptops will soon be able to support virtual reality. When this happens, there will be an explosion of training and coaching virtual reality products aimed at both businesses and individual learners.

Virtual reality agile training has sharply different economics than face-to-face training. A live Scrum course may have 20 students which have paid $1K each, producing gross revenues of $20K. We believe that agile training in virtual reality will be able to comfortably 'hold' spaces for hundreds of simultaneous learners who will then be able to pay approximately $200 each for the learning experience, thus sharply changing the economics of training and creating a 'winner takes all' race in which the trainers who are first to develop and embrace this technology will dominate the market. The winning trainers will be able to invest hundreds of thousands or even millions of dollars in creating virtual reality environments that support learning, far outspending what face-to-face trainers can afford.

## Case study

One of us (Elena) has used a 360 degree camera (Orah 4i, Nikon KeyMission 360) in coaching and training situations as a way to explore how virtual reality and online technology will disrupt the agile space.

As agile coaches we are always looking for the opportunity to coach developers, product owners, and managers. However, it is hard for people to reach out to us and ask for coaching, and it is even harder for us to reach out and to offer coaching because this is often received as an implied criticism (people often think that it means that they are not performing or a manager isn't happy with their work results).

To address this situation in a supportive and non-confrontational manner, VR recordings of a team's Scrum ceremonies and Agile training classes were made. These recordings provide an external point of view where a viewer can choose what he/she would like to observe. The first VR video was of the team's Scrum Retrospective. With VR goggles an individual viewer can see what's going on in the room from different 360 degree points of view and move the attention to different objects, discussions and people. The viewer can also observe their own behavior and body language during the group discussions. Shortly after viewing the video quite a few individuals reached out and asked to be coached on the Scrum ceremonies and the team's collaboration activities.

VR video allows individual viewers to get a neutral external point of view of their choice. VR allows the viewer to observe not just one's own behavior during the team activities and discussion but also the team's reaction to it. The small size of the VR camera and 360 recording angle doesn't disturb a group discussion and doesn't disrupt an individual because "they are on camera". Nobody knows what the viewer might choose to see in the recording.

As the next step in this coaching approach, similar situations to the ones observed in the video were staged so that the coachees could once again observe and experience their behavior. These stage simulations are similar to an Escape room exercise where participants don't know what to expect next. After a few recordings of these immersive situations, the coachees said that the VR viewing increased awareness of their own role in the team's activities and discussions. They also said that Product Owners started to use new collaboration and communication tools and techniques and found the coaching to be very helpful in the daily work with the teams.

The total cost of the hardware and software was approximately $700. Given the low cost and the relatively little skill needed to operate the equipment, we expect this practice to grow.

## What will happen

We believe that online technology and virtual reality in particular will have a disruptive effect on agile online technology and coaching in the coming three to five years.

We make the following predictions with 90% confidence:

- Prediction: By June 30, 2019 thousands of people will have participated in agile activities in virtual reality (e.g., story mapping or an online conference).
- Prediction: By the end of 2019 thousands of people will have taken an online agile course which they consider to be excellent.
- Prediction: By the end of 2020, at least one large, credible agile/Scrum certification organization will be running agile/Scrum certification courses in virtual reality.

Adoption of online technology is lagging in the agile space in part because one of the Manifesto principles is, "The most efficient and effective method of conveying information to and within a development team is face-to-face conversation." The agile training community is highly biased towards face-to-face training. This allows online training technology to focus in related fields, such as software development and design, so when it finally enters the agile space it will already be a mature technology.

If you are currently working in the agile space, we urge you to prepare for this tectonic shift by upgrading your online technology skills. Consider learning how to use online facilitation tools (such as IdeaBoardz), participating in online agile user groups (such as the Agile Practitioners Online Special Interest Group), and experiencing virtual reality and augmented reality.

***

To read this article online with embedded hypertext links, go here:

**https://www.infoq.com/articles/virtual-reality-disrupt-coaching/**

# About Michael de la Maza

**Michael de la Maza** is a Scrum Alliance Certified Enterprise Coach (CEC). As an Agile consultant, his major engagements have been with Paypal, State Street, edX, Carbonite, Unum, and Symantec. Previously, he was VP of Corporate Strategy at Softricity (acquired by Microsoft in 2006) and co-founder of Inquira (acquired by Oracle in 2011). He is the co-author of Professional Scrum with TFS 2010 and Why Agile Works: The Values Behind The Results. He holds a PhD in Computer Science from MIT. He is on email at michael.delamaza@gmail.com and on Twitter @hearthealthyscr.

# About Elena Vasilieva

**Elena Vassilieva** is Scrum Alliance Certified Team Coach (CTC). She helped the teams at Warner Brothers, Twitter, Best Buy, StubHub, IBM and other great companies to bring together Design Thinking, Agile and Lean approaches into the Product development. Elena has a background in Engineering management and she is a Certified Scrum Professional. Elena  is shooting VR documentaries and uses online video streaming to create an immersive coaching experience. Her e-mail is havasupai3@gmail.com.

# Why is psychological safety being ignored?

By John Dobbin

Posted on November 1, 2018

El Greco (Domenikos Theotokopoulos) - Laocoön

*In one of the most extensive business research projects ever conducted, Project Aristotle, Google found, to their surprise, that the number one driver of team performance is psychological safety.*

This is a gift on a plate to organisations. Want to radically improve your performance? Then simply make psychological safety a priority.

The response ... crickets. My partner, Dr. Richard Claydon, and I talk to a lot of organisations about this. You'd expect it to be top of most senior executive's agendas. But, alas, it is not. Nowhere near it. Following are some of the reasons why, in no particular order.

## Little or no Understanding of Psychological Safety

Many executives simply don't know about it. They haven't read Laura Delizonna's article in Harvard Business Review, they haven't seen Amy Edmondson's TED Salon talk on it, they missed the New York Times special feature, and they haven't read Google's guidebooks on team effectiveness or any number of other articles on psychological safety. Ignorance is bliss; but not an excuse.

## It's Misperceived and Misunderstood

A number of articles have confused psychological safety with safe spaces and low accountability. Safe spaces are places in which you go if you do not want to be confronted by ideas and opinions that are offensive to you. In psychologically safe environments, you need to speak up. Although Amy Edmondson does identify a high psychological safety / low accountability environment (she calls it the comfort zone), she makes clear that psychological safety works in parallel with accountability. If there's high psychological safety and high accountability, it's a learning zone. By the way, low psychological safety and high accountability? That's the anxiety zone.

## "It's a fad"

Fads come and go all the time in business, and one should remain healthily skeptical. However, the depth of Project Aristotle's research cannot be ignored: 2 years, 180 teams, double-blind interviews, and over 35 statistical models analyzing hundreds of variables. The data and insights cannot be ignored by anyone who is serious about business performance.

### "It's not for us"

Could be true. Amy Edmondson states that high psychological safety is only important in landscapes of uncertainty and interdependency. For work which is straight-forward and can be done without relying on others then business-as-usual might be fine. Or even better, replaceable by an AI.

### Senior people don't want this spotlight on them

Quote from the HR department of a listed company about running a workshop on psychological safety: "We can't put this forward. There are people at the top that will feel extremely threatened by this". So a public company can not partake in profit-generating skill development because it may expose the toxicity of senior executives. Herein lies a systemic problem. To paraphrase, "we know we are doing a bad thing but we don't want to look at it, don't want to expose it, and don't want to treat it." It is not just one organisation saying this, we get variations of the same sentiment a lot. An awful lot.

This is a board-level problem. Simply put, if a CEO is not committed to developing a company-wide competence that will very likely improve organisational performance, then one that does should be found.

### Superficial treatment

"7 (or 10, or 14) ways to create psychological safety in the workplace", posted on the intranet, is not a solution. Organisations that have normalised toxicity have to reverse it. It will take time. It will require dedication and perseverance. It will produce casualties. But it will definitely produce results.

Inculcating psychological safety has a raft of obvious benefits:

- People, whom companies spend a fortune on for their skills and experience, will actually contribute to collective intelligence
- Fear will be replaced with Seeking and generate better thinking, collaboration and innovation
- Good people will stay, while those who rely on aggression will depart
- The workplace will become more vibrant and alive, people will enjoy their work more
- Employees won't take as much stress and anxiety home with them, they will become physically and mentally healthier

### How to get started:

1. Understand what it is and how to foster it. Read up. Go to courses.
2. Start with any team. Get buy-in. Enlist internal influencers. Spread the knowledge via the social network of the organisation, face to face, action by action (we refer to this as "preparing the soil").
3. Identify blockers, those people who actively undermine safety. Either convert them, work around them, or get rid of them.

### References

Delizonna, L, 2107, 'High-Performing Teams Need Psychological Safety. Here’s How to Create It', *Harvard Business Review*, https://hbr.org/2017/08/high-performing-teams-need-psychological-safety-heres-how-to-create-it

Duhigg, C, 2016, 'What Google Learned From Its Quest to Build the Perfect Team', *New York Times*, https://www.nytimes.com/2016/02/28/magazine/what-google-learned-from-its-quest-to-build-the-perfect-team.html?smid=pl-share

Edmonson, A, 2018, 'How to Turn a Group of Strangers into a Team', *TED*, https://www.ted.com/talks/amy_ edmondson_how_to_turn_a_group_of_strangers_into_a_team

Edmondson, A, 2018, 'The Fearless Organization: Creating Psychological Safety in the Workplace for Learning, Innovation, and Growth', *Wiley*, https://www.wiley.com/en-au/The+Fearless+Organization:+Creating+Psychological+Safety+in+the+Workplace+for+Learning,+Innovation,+and+Growth-p-9781119477266

Google, 'Guide: Understand team effectiveness'; *Google Re:Work*, https://rework.withgoogle.com/print/guides/5721312655835136/

***

To read this article online with embedded hypertext links, go here:

**https://www.linkedin.com/pulse/why-psychological-safety-being-ignored-john-dobbin/**

# About John Dobbin

**John** Dobbin is a seasoned technologist, business leader and consultant. His background is in mathematics, computer science and organisational development. As the Founder and Principal of a prominent systems integration firm in the early 2000's, he was an early adopter of Agile and assisted numerous multinationals make the transition to agile delivery. Today he helps organisations all over the world execute digital and agile transformations.

John continues to research and explore the frontiers of behavioral science and complexity theory within the context of organisational development, developing new models and materials to assist agile practitioners think and work more effectively.

# Understanding Agile Metrics

By Bob Ellis

Posted on October 11, 2018

An Agile transformation is intended to build new business capabilities- to get things done faster, to improve product quality, to reduce risk, to improve predictability, and to improve productivity. Yet, many Agile transformations lack any plan for measuring success. Executives supporting Agile transformation programs often become frustrated with the lack of meaningful metrics. Meanwhile, practitioners debate the relevancy of metrics, often from a silo rather than an end-to-end perspective. Months into the transformation, executives wonder whether the program even merits continued funding. They struggle to pull together data after-the-fact. Because they didn't have baseline measurements from the start of the transformation, it is hard to demonstrate progress. Often, any data that is freely available becomes the metric of focus.

### So, how can you get started on the right track to measuring and proving the success of your Agile transformation?

The primary outcomes of Agile transformations are apparent across three dimensions:

- **Flow**: Building value incrementally and quickly. This enables prioritized delivery and feedback, and highlights impediments.
- **Value**: Building the right thing. This prioritizes "outcomes over outputs," and delivers more with less.
- **Quality**: Building the thing right. "Built-in-quality" reduces rework and drives productivity.

Alignment across these three dimensions is critical, as is the alignment across all levels of the organization, from team and program, all the way to portfolio management and strategy. While Agile derives speed and innovation through the decentralization of decisions, some decisions must be centralized for the sake of alignment: strategy, vision, road map, and success measures across technology, product, and flow. The measurement of the abstract concept of the successful alignment across these platforms can then be measured with tangible data.

## FLOW: Building it Fast

"Velocity" is the most common metric, and it is the one most often abused beyond the Agile team. In other cases, "feature percent complete" based on linked stories can also provide similarly misleading information rarely worth the investment it takes to assemble.

A good measure is the "cumulative flow diagram" (CFD) of stories moving through states, which provides visibility on cycle time, work-in-progress, velocity, and an estimated time required to complete the overall plan. A cumulative flow diagram provides good feedback on flow and lack of flow, which is caused by impediments. The cycle time of a user story from "in-progress" to "accepted" is another measure of flow. Six sigma practices can be used to identify and reduce variation in cycle times and increase flow accordingly. Cumulative flow for features is another important metric that most often highlights cross-team dependencies owning different components needed for delivery of a feature. Cross-team dependency and solution integration impediments, highlighted by feature (or larger work item) cumulative flow, can often be resolved by executives that own the end-to-end value delivery.

The best flow metrics will also include an element of employee engagement or employee happiness. When this metric is trending the wrong way, soft skills and team emotional intelligence are needed to understand and resolve any contributing issues. In this case, an examination of outliers often reveals the root cause of the real problem, rather than a focus on averages.

> **Who is responsible for *flow* in an Agile transformation?** The scrum master at the team level and those in the facilitation role as we go up the organization, all the way to facilitation of strategy definition, Agile program

management, and governance. In SAFe, these are the Release Train Engineers, the Solution Train Engineers, and the Portfolio Management Team. In the Scrum at Scale Framework, this is the Executive Action Team (EAT).

## VALUE: Building the Right Thing

Planning and reviewing, including a demonstration of working software, helps to manage the risk involved in execution and closes the "credibility gap" between IT and business. The frequency of this business and IT collaboration is the primary indicator of whether the right thing is being built. Is there clear communication regarding vision, a road map, a prioritized project backlog, and the purpose of the organization? Is there collaboration and alignment on business value and the highest value release plan? Is there opportunity to review a draft of the release plan and an opportunity for the teams delivering the work to estimate and propose a final release plan? Does the culture validate the hypothesis and assumptions with data from larger programs before going all-in on large investments?

A good practice for reinforcing collaboration and the measurement of distinct, relative business value is the Program (value) Predictability Measure from SAFe.1 Program-Value Predictability involves sponsoring executives assigning a "business value score" during planning meetings for each longer-term business objective. This practice provides the best proxy for business value - it is collaborated by business owners and it is done at the latest-possible responsible moment. It is relative to the ratings of other business objectives, and it is used again upon delivery to provide a plan vs actual score, which even accounts for changing business value between the plan date and deliver date. This SAFe metric uses the same basic construct as the Objectives and Key Results (OKR) measure from Silicon Valley and Intel.2

**Who is responsible for *value* in an Agile transformation?** On the team level, the Product Owner (PO), higher-level Product Manager, and higher-level business people. These people need to be aligned with the sponsor(s) of the delivery organization. These can be owners of an operational value delivery stream such as customer service or a product and marketing organization, or VPs and executives of the company. They must make decisions that maximize short and long-term value delivery.

## QUALITY: Building the Thing Right

Agile implemented properly has a pre-defined quality gate for each team, with the PO serving as a check and balance to accept stories and ensure that definition is followed. Demonstrated capacity is understood through many data points of actual delivery. Additional quality gates downstream are in place to ensure the product is of customer quality before releasing to production. All the downstream, post-story-acceptance work is "technical debt," which includes bug fixes, running tests, fixing tests, release procedures, change control, including customer feedback, and customer service costs. The ultimate vision is built-in -quality and complete elimination of technical debt. This is only possible through a culture of collective ownership across Development and Operations.

A good quality metric is a count of defects post story-acceptance. A leading indicator of improved quality includes the calculation of automated unit test coverage, feature test coverage, and non-functional test coverage. A good practice is that downstream defects are not only fixed, but also automated tests are promoted to earlier layers in the test harness so that cycle time to detect such defects is shorter. The best predictive measures of quality improvement are investment in telemetry of the integration and deployment pipeline. Tools in this space highlight the location of major bottlenecks and problem areas, so that the development organization can target the most important area to improve next.

**Who is responsible for *quality* in an Agile organization?** On the team level, the development team, the higher-level system or solution architect, and the enterprise architect. Alignment on the technology strategy of the overall organization provides guidance on emerging design formed by the development team that is doing the work.

## SUMMARY

Ironically, each of the best-practice Agile metrics are a measure of "flow of business value." Short feedback loops are used to improve flow, value, and productivity. Many documented case studies indicate these are the best metrics to guide successful management of a complex business system in a fast-changing environment.

Want to learn more about how to measure your Agile success?

See Our Agile Resources

---

1. “Metrics.” Scaled Agile Framework, 2010-2018, http://scaledagileframework.com/metrics/#T4. Accessed 11 Oct. 2018.
2. “The Beginner’s Guide to OKR.” Felipe Castro, https://felipecastro.com/en/okr/what-is-okr/. Accessed 11 Oct. 2018.

***

To read this article online with embedded hypertext links, go here:

**https://blog.eliassen.com/understanding-agile-metrics**

# About Bob Ellis

After 3 years as a developer of medical imaging products for HP, **Bob** broadened his career over 3 decades ago as a lean management consultant with PWC and focus on cost, quality and delivery for clients across multiple industries.

Bob Joined EMC for 20 years and initially managed manufacturing, supply chain, inventory management and built IT Systems that enabled 50X market cap gain. As manager, director and internal consultant: Bob worked every functional area and culminated as Value Delivery Architect for 1500-person product line which consisted of hardware, firmware and software for $6B VMAX data center infrastructure product line.

Bob now works for Eliassen Group as Agile Transformation Advisor, keeps “in the trenches” as coach at all levels, helps clients plan and execute practical improvements to value delivery and success metrics.

Bob’s degrees are Cornell BS Engineering and MIT MS Management: Investments and Systems Engineering.

blog.eliassen.com

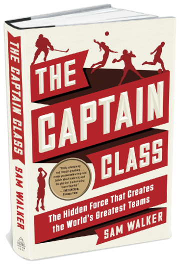
# Leadership Lessons for Creating High-Performing Scrum Teams

**After analyzing the traits that make a great team captain, we take a look at how these qualities apply to Scrum leadership.**

By Ron Eringa

Posted on September 28, 2018

This is the last in a series of 3 blogs presenting the result of an interesting research study from Sam Walker.



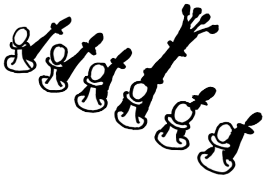
Walker discovered that the most successful sports teams that ever existed all shared one single element: they all had a team captain with 7 overlapping traits that made them extremely successful. In this blog, we will explore what Agile Leaders can learn from these extremely successful team captains.

## 6 Lessons to Learn from Elite Team Captains

The Scrum Master role has a lot of overlap with the team captains from Walker's research.
Another overlap with Walker's research is the role of the sports coach: the Agile Leader, responsible for the Scrum Teams.

So, what lessons can Agile Leaders or Scrum Masters learn from these sports teams?

### Lesson 1: Scrum Masters Make the Difference Between Good and Great Teams

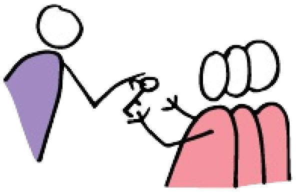


Walker's research proves that having an inside Leader (or team captain) is the most important factor in making a team successful.
It wasn't strategy, management, money, or superstar talent that made the difference: 106 Teams had similar characteristics but all ended second place (the so-called Tier 2 teams). It was the presence of a Leader, fighting on the battlefield with the team, who made the difference between good and great.

In Scrum, it is the Scrum Master role who has most overlap with the captains from Walker's research.

In times of high pressure and when things get rough, the Scrum Master is the leader, working with the team from the trenches. For this reason, he will have the most impact on the teams' performance. No coach, manager or process can help a team better in these circumstances than the Scrum Master.

### Lesson 2: Agile Leaders Enable Scrum Masters to Lead



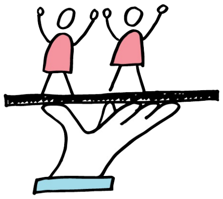
Many Scrum Masters I encounter, only take up day-to-day routine tasks (such as hosting Scrum events and planning meetings), because their manager handles all leadership-related work.

The Tier 1 captains in Walker's research proved that their Leadership was most effective because they were part of the team. The duty of their coach was to create an environment that allowed this to happen.

An Agile Leader should help Scrum Masters to develop the character traits of a Tier 1 captain, so Scrum Masters can also become Agile Leaders. While some Scrum Masters might have a natural talent to lead, some will develop these talents along the way. The challenge for the Agile Leader is to understand when to delegate these responsibilities, once the Scrum Master becomes more mature.

Alex Ferguson, the legendary coach of Manchester United once said: "As hard as I worked on my own leadership skills, and as much as I tried to influence every aspect of United's success on the field, at kickoff on match day things moved beyond my control."

### Lesson 3: Scrum Masters are Servant Leaders



We have come to expect that the best leaders are often those with:

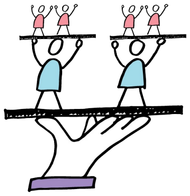
- Exceptional talent
- Mesmerizing characters
- High market-value
- Superstar egos

The evidence Walker presents (the best team captains are Servant Leaders), proves that this is a distorted picture.

Many Scrum implementations I have seen reflect that same distortion. We often think that Scrum Masters are highly technical superheroes that follow orders from an even greater hero-leader. As a result Scrum Masters are often selected for their technical skills and their superhero status.
The most effective Scrum Masters I encountered had the character traits that Walker found in his research.

Walker's research proves that Scrum Masters do not have to be superstar heroes with deep technical skills. Instead, they should be humble, have a relentless drive to learn and play to win. While doing this, they should support team members in growing and becoming technical experts.

### Lesson 4: Agile Leaders Are Also Servant Leaders



Walker could not find any evidence that the sports coaches (the equivalent of the Agile Leader) of the Tier 1 teams had a direct impact on their success. With regard to the contribution of the coach, his research led Walker to a number of conclusions:

- The coaches were not prizewinning strategists
- Most were not inspirational figures
- The coaches did not have a big impact on a player's performance
- Changing coaches had either no or a low impact (many of the Tier 1 teams coaches came and left during the bursts of success)
- There were no real unifying principles/character traits like with the teams' captains

What these coaches did have in common:

- They gave their captains the room to be a leader for their team
- They all enjoyed close and contentious relationships with their captains
- They all had been decorated captains before moving to their management position

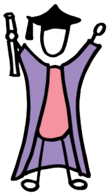
This allowed them to understand what makes a good captain and identify the perfect person to lead the players.

All coaches of the Tier 1 teams understood that to achieve great success, they needed a player on the field who could serve as their proxy.

The consequence of delegating Leadership to a Scrum Master is that the role of the traditional manager will change. The focus of an Agile Leader might be different, but the roles and required skills/character traits have a lot of similarities.

True Agile Leaders are also Servant Leaders. They work closely together with the Scrum Masters and dare to step aside, once the game is on.

## Lesson 5: Agile Leaders Create a Learning Environment

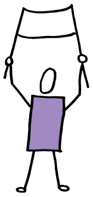


Walker's research shows us that being a great leader is not (always) genetically determined, it can be learned. All Tier 1 captains started off with having early struggles in their captaincy. These struggles lead to a breakthrough moment that left no doubt about their desire to win. As a result they focused, worked hard and pushed themselves and others to keep learning.

Coaches and managers of Agile teams need to create an environment where it's safe to fail and where people get enough opportunity to learn from their mistakes.

It is the Scrum Master's job to show the team how to be persistent in reaching their goals and learn from their mistakes.

### Lesson 6: Positive Dissent Is Essential



Scrum uses closed feedback loops as a mechanism to continuously improve, eliminate waste and create incremental value. All Tier 1 captains from Walker's research understood that their teams needed positive dissent in order to make such a feedback loop work.

Like the captains in Tier 1, a Scrum Master needs to create an atmosphere in his team where conflicts are not driven by egos, but by the will to win.
To achieve such an atmosphere, a Scrum Master needs to act on the edge of the status quo.

To be excellent, it is sometimes necessary to challenge existing processes, bad decisions, or test the limits of existing rules.

## Conclusion

One of the biggest struggles in becoming a true Agile Leader is to delegate responsibility. Walker's research shows that Leaders who are able to delegate these responsibilities to their Scrum Masters, create the most successful teams. If the Agile Leader and the Scrum Master work closely together, they can achieve amazing things!

Did you become curious about the role of the Agile leader? Come and experience it in my Professional Agile Leadership (PAL-E) training!

***

To read this article online with embedded hypertext links, go here:

**https://dzone.com/articles/leadership-lessons-for-creating-high-performing-sc**

# About Ron Eringa

**Ron** Eringa inspires Leaders to develop a culture where Agile teams can be successful. He has more than 18 years experience with leadership development in IT-driven organizations. Ron is a public speaker, Professional Scrum Trainer and writes a popular blog on the evolution of Agile organizations. As curriculum steward, Ron is also responsible for the development of the Professional Agile Leadership courses at Scrum.org.

Ron strongly believes that this complex digital age needs a new kind of leadership. He supports managers, executives and teams to develop a leadership style that enables a culture of creativity, ownership and engagement in their organization. With Culture Design workshops, Leadership trainings and the Scrum.org training classes he helps organizations to develop learning journeys for agile teams and leaders. Ron recently started a new company called the Agile Leadership School to provide online learning journeys for Agile professionals.

More info: http://roneringa.com/

# What All Great Teams Practice

By Mackenize Fogelson

Posted on October 26, 2018

“The way we do **anything** is the way we do **everything**.”

MARTHA BECK

@MACKFOGELSON
THE READY

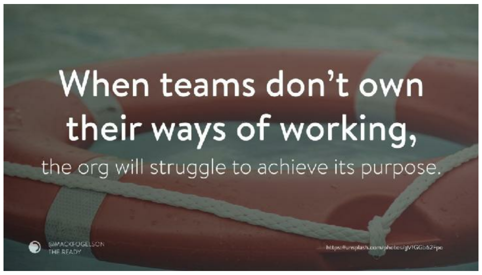
**Ways of doing** are the actions your team and organization operationalize so you can work in new, adaptive ways. Ways of doing include things that change your systems — how you budget, plan, allocate resources, determine compensation, or give feedback, for example. It can also be things like how you meet, communicate, prioritize, craft your vision, or develop and disseminate strategy.

**Ways of being** are the mindsets and behaviors that are integral to changing your ways of doing across the organization. It’s the individual humans in your company shedding habits and patterns (like retaliation, blaming, and shaming) that don’t serve them. It’s trading those less desirable ways of being for a new lens (like coming from a place of curiosity or having empathy) that helps nurture relationships with the other humans we work with and manage the complexity of our environment.

Successful cultural transformation requires both.

I’ve spent most of my career studying what connects people to the companies they work for and the brands they choose to evangelize and support. *When purpose is integrated into an organization’s operating strategy — along with customer and employee insight — it plays an integral role in an organization’s growth.*

Purpose can be as big as something that leads the vision of a 25,000 person organization, and also as small as what guides the many 15 person teams inside that same org. As big as purpose can seem, it’s really in the small moves that small teams make that adds up to accomplishing really big things as a company.



**When teams don’t own their ways of working,**

the org will struggle to achieve its purpose.

@MACKFOGELSON
THE READY

https://unsplash.com/photos/gVfGGb62Fpc

What I have also learned is that it doesn’t matter how inspirational, grand, or appealing a company’s purpose is, if their teams are not given the opportunity to take ownership for their ways of working, the org will struggle to achieve its purpose. Not only that, but you’ll have a whole lot of miserable, unproductive humans working on stuff that doesn't really take the company anywhere.

Ownership requires curiosity;
learning to let go of old ways
while being open to practicing new ways.
@MACKFOGELSON
THE READY
https://unsplash.com/photos/5IHz5WhosQE

In my daily work, I guide executives, their teams, and their teams of teams to design — and then scale — new ways of working. Over time, this work becomes a new operating system for the entire organization. What's powerful for these teams is they are given the decision rights and the voice to contribute to the company's purpose every day.

But in the process of that transformation, there is a great deal of friction they must wrestle with because of what has been deep-rooted in their existing system and culture.

## Learned helplessness

Executives and managers struggle to handle the amount of complexity they're faced with. Many times — instead of trusting their people — they shift into this place of always needing to have the answer. Of controlling everything around them — including what everyone does.

Rather than being trusted to steer towards the company's purpose and vision, people and teams get used to waiting around to be told what their strategy and priorities are. Across the org, people become accustomed to being told what they can and cannot do. Over time, they just give up their initiative. They learn to follow directions and obey the check-list and follow the rules and do what they are told. Slowly, all of their agency, creativity, and insight escapes into the ether.

## Retaliation

Along with learned helplessness, I've also observed a great deal of **retaliation**. This shows up when — time and time again — policies, bureaucracy, rules, regulations, and the compliance meant to control the complexity suck the culture of experimentation right out of people and teams.

People get tired of getting in trouble, so when things are learned from projects that don't work out, rather than sharing and opening up conversations and making that learning visible, people keep their mistakes to themselves, avoiding any potential for retaliation.

## Shame

A learned helplessness and retaliation culture also breeds other not so awesome things like a whole lot of **shame** and **judgement** — people projecting this on themselves and those they work with which produces **blame**.

## Blame

Blame becomes the new flavor of accountability. When a team has a project to lead, designating a leader is not so much about who can kick ass in leading through the work, it's about who is going to be blamed when the efforts go south. People learn to associate change with something that is done by everything and everyone outside of themselves, failing to remember they always have a choice.

## Fear

And all of these patterns really just stack up to a culture that operates with a whole lot of **fear**. Fear that gets in the way of being honest with ourselves and others. Deep, dark fear that discourages all of the risk taking. All of the ingenuity. All of the possibility and light. All of the good stuff we were hoping to get from the people we hired and trust to do their jobs.

Orgs keep adding stuff
like agile, scrum, and lean
but keep missing the important stuff in between.

@MACKFOGELSON
THE READY

https://unsplash.com/photos/r2jpr8MDw0I

But organizations want to be faster, better, smarter, and more innovative, so they bring in shiny new tools and technology. They introduce methodologies like agile, scrum, six sigma, lean — you name it — thinking, if we just do these things the right way we will have all the stuff to achieve world domination.

And although these methodologies can be useful and serve a purpose, organizations are missing some really important stuff that is getting in the way of how their organization operates at its fullest.

Teach your teams
to manage complexity
*and* new ways of *being* with each other.

@MACKFOGELSON
THE READY

https://unsplash.com/photos/kvIAk3J_Ajo

Teaching teams how to be adaptive and manage complexity is just one part of this equation. Organizations also need to teach people how to be with each other. How to be brave and vulnerable and honest. How to have healthy conflict. How to communicate with each other. How to fall down and get back up. This gives teams the power and opportunity to own their ways of working which leads to achieving your organization's purpose.

When teaching organizations and their teams new ways of doing and being, resistance always plays a part. Even those brave souls who are eager to learn new ways and shed the old, they can't help but gravitate towards the stuff that appears easy; like **team chartering.**

> "Let's get everyone on the team in a room together and clarify our mission, purpose, and values. Let's spend the time to name how we're going to communicate and the technology we're going to use to do our work. Let's open up a dialogue about the meetings that will shape our operating rhythm and identify some guardrails and norms that give us some autonomy. Let's get crazy and even get into roles, responsibilities, and decision rights."

Don't get me wrong, this is all important stuff. There's just a few things to consider before diving into the components of your team charter.

**Addressing the emotional side** before you team charter will save you from a shit show. I have yet to experience a team that didn't have demons lurking beneath the surface. Even if a team is predominately healthy, there is always baggage to be considered.

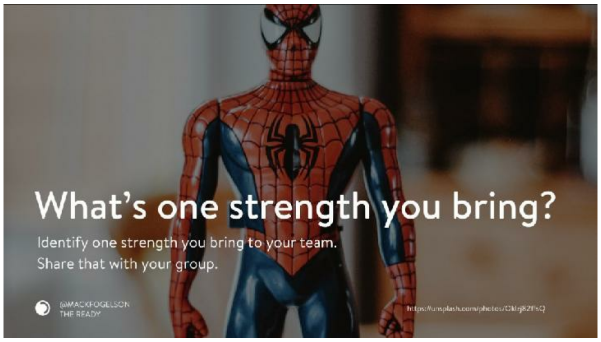
Regardless of the history of your team, start with the stuff that makes them (and you) uncomfortable. Start with the stuff that will make you feel. Open up the space for emotions and real conversations to be had. Having the courage to take your team here first will pay dividends. You can then work towards the components of your team charter (all you need to do that is below).

## Ask questions and make space

Especially if your team is in a really dysfunctional place, you will want to bring in a trained facilitator to open up and hold the space for the hard stuff to surface and determine the best next steps. Keep in mind you could be looking at several sessions with your team before jumping into identifying and clarifying the components of your team charter. Take it slow.

With the first session, allow up to three hours. If your team is willing to drop in and be vulnerable with each other quickly, you'll have a lot to talk about.

One at a time, ask these questions:

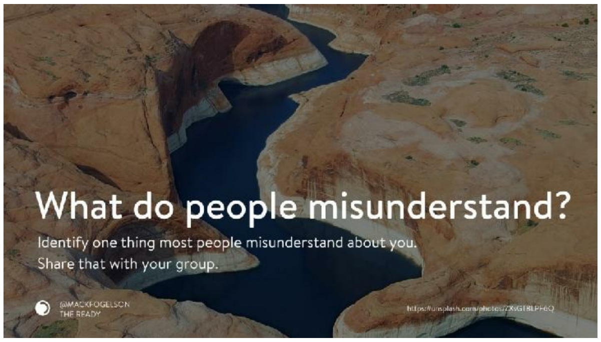


What's one strength you bring?

Identify one strength you bring to your team.
Share that with your group.

@MACKFOGELSON
THE READY

https://unsplash.com/photos/Cklzj82FsQ

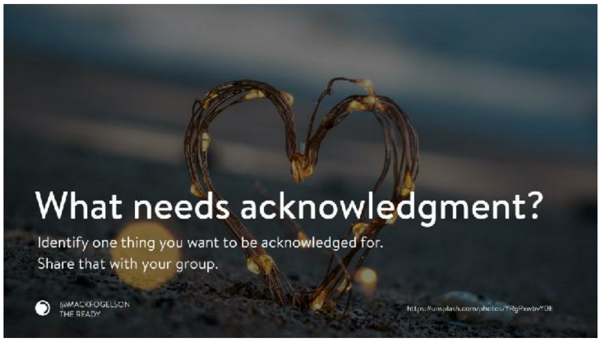


What do people misunderstand?

Identify one thing most people misunderstand about you.
Share that with your group.

@MACKFOGELSON
THE READY

https://unsplash.com/photos/ZXsGTBLPF6Q

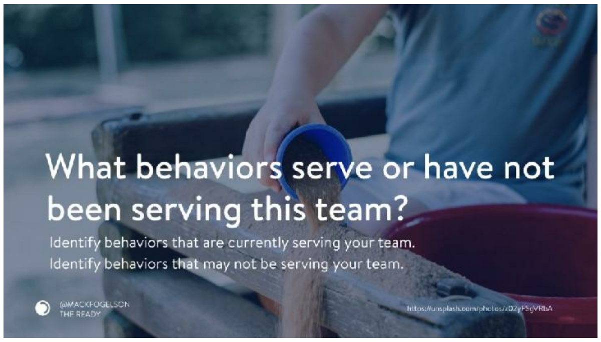


What needs acknowledgment?

Identify one thing you want to be acknowledged for.
Share that with your group.

@MACKFOGELSON
THE READY

https://unsplash.com/photos/YRgPxwbvY0E



What behaviors serve or have not been serving this team?

Identify behaviors that are currently serving your team.
Identify behaviors that may not be serving your team.

@MACKFOGELSON
THE READY

https://unsplash.com/photos/zD2yFSgVRbA

With each question:

1. **Give time to self-reflect**
   Allow silent time for introspection. Have each team member jot down their answers down on a piece of paper or a sticky note.
2. **Share out**
   Ask the team to share their responses in rounds (only one person speaks at a time). Hold space for each person to be heard.
3. **Listen**
   Have the facilitator listen for values, guiding principles, guardrails, and norms. This is a great opportunity for the truly authentic components of what’s important to your team to surface even without your team intentionally working on them for your charter.

Typically the struggles teams face come down to not having — or not knowing how to have — hard conversations. You have to make the space for it.

Remember when your team is not performing

they need some space to name stuff.

@MACKFOGELSON
THE READY

https://unsplash.com/photos/cGXdjyP6-NU

When your team starts to bring up and work through their ghosts, encourage them to give specific examples. Take all the time required to hold space for what your team needs to surface before working through the components of your team charter.

Once you’ve addressed some of the deeper wounds that may be present in your team, you can begin to introduce some ways of being that will become part of how they work every day. This is not the soft stuff. Ways of being will always be the hardest work they will ever do.

Self-awareness

@MACKFOGELSON
THE READY

https://unsplash.com/photos/fUqN6nRc1jc

A decdication to cultivating self-awareness is one of the best gifts you can give your team as it is what opens up the space for your team to do great work.

There are many ways and tools that will help your team practice self-awareness. One of the simplest is called a check-in. Check-ins are different than one-on-ones or stand-ups or huddles with your team. Protect the first 5 minutes of every meeting and make the space to hear from all of your humans.

Check-ins can add levity and authenticity to the room

which opens up the space to be human.

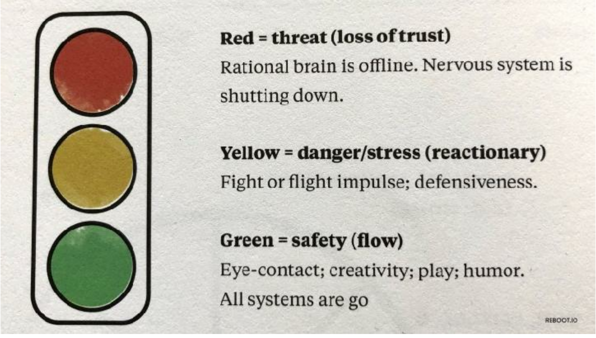
@MACKFOGELSON
THE READY

https://unsplash.com/photos/8UcNYpynFLU

Check-in questions can be simple like: What has your attention? What's your favorite food? If you were marooned on an island, what album would you want to listen to?

Start there.

When your team is ready to practice vulnerability, try mixing in the Red/Yellow/Green — a check-in practice I learned from Reboot.



**Red = threat (loss of trust)**
Rational brain is offline. Nervous system is shutting down.

**Yellow = danger/stress (reactionary)**
Fight or flight impulse; defensiveness.

**Green = safety (flow)**
Eye-contact; creativity; play; humor.
All systems are go

REBOOT.IO

This type of a check-in helps your team build their self-awareness muscle. After you've checked in like this for a few weeks, ask your team to reflect on what they're noticing. Are they checking in *green* every time and really present, or maybe not naming something that's really there? When they are *yellow*, where are they feeling it in their body? How many times are they coming to check-in with *red*?

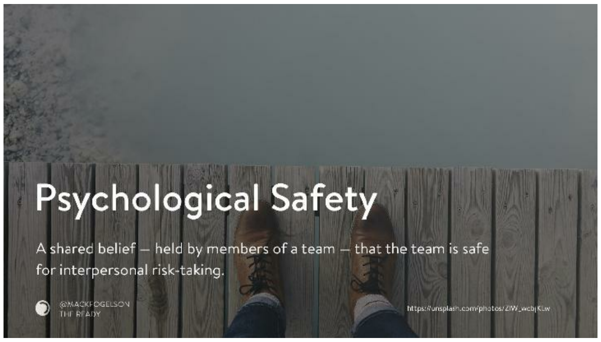
What did you notice?

@MACKFOGELSON
THE READY

https://unsplash.com/photos/hpZh7Pvs9mk

Check-ins are not just for connecting, they are to practice noticing. Keep asking your team the question: What did you notice?

Over time they will have a much better sense of how they’re truly showing up with their team and what they may want to try doing differently.



Psychological Safety

A shared belief — held by members of a team — that the team is safe for interpersonal risk-taking.

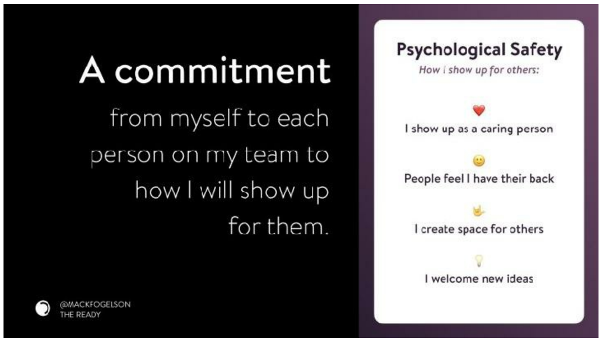
@MACKFOGELSON
THE READY

https://unsplash.com/photos/ZIW_wcbjKLw

Self-awareness is a prerequisite for building psychological safety. Introduced by Amy Edmondson and studied by Google, it has been found that psychological safety is the key component to successful teams; a shared belief that when working on a team, it is safe to be vulnerable with each other and take interpersonal risks.

A small move you can make with your team on fostering psychological safety is with a daily reminder. We have had great success spreading this across a culture with small cards that briefly state a commitment to the practice. We read these at the start of every meeting.

There are two sides to these cards. The front — which communicates the commitment each individual member of the team is personally making to how they will show up for the other members of their team.



A commitment

from myself to each person on my team to how I will show up for them.

@MACKFOGELSON
THE READY

Psychological Safety

*How I show up for others:*

❤️
I show up as a caring person

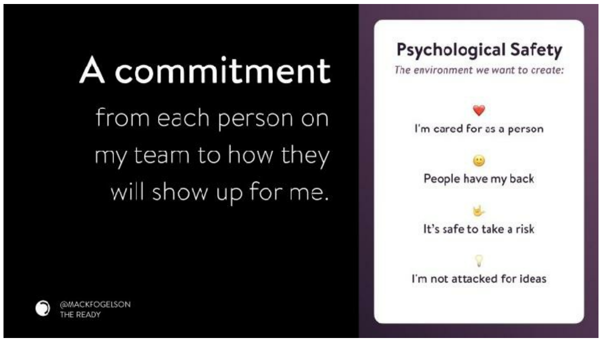
🙂
People feel I have their back

🤟
I create space for others

💡
I welcome new ideas

And the back — which communicates the environment the entire team as a whole is committing to cultivating.

Psychological safety begs the questions: Can I speak up in a meeting? Can I share my ideas? Can I contribute at work and not be penalized or have something I say used against me?



A commitment

from each person on my team to how they will show up for me.

Psychological Safety

*The environment we want to create:*

❤️ I'm cared for as a person

😬 People have my back

🤘 It's safe to take a risk

💡 I'm not attacked for ideas

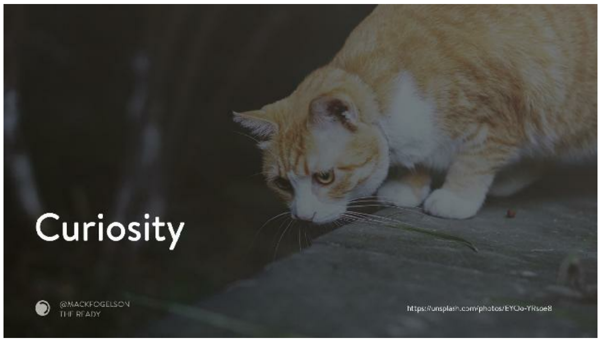
@MACKFOGELSON
THE READY

It means the individuals on your team can feel safe to be honest, share ideas, and show up as who they really are and they will be supported. Even when there are things that need to improve.

Psychological safety does not mean you don't have conflict or difficult conversations with the people you work with. The longer your team is together, the more opportunities will present themselves for your team to both build and destroy trust and psychological safety. That's part of being human and the process of better understanding yourself and the effects actions have on your team.

What's important is to continually make the space to not only practice but also look underneath the surface and get curious about what's really going on with your team when you sense things are off.

Especially when a team is dysfunctional or has toxic behavior going on, it can take a really long time to build this type of safety. Continue to make space for it.

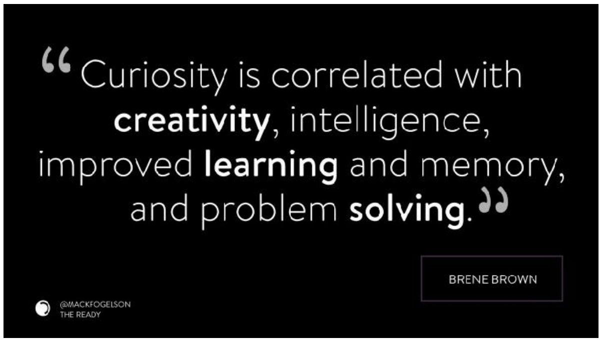


Curiosity

@MACKFOGELSON
THE READY

https://unsplash.com/photos/EYOo-YRsoe8

As your team explores new ways of being, self-awareness and psychological safety are two really important things to practice. Another big one is teaching the art of curiosity.



"Curiosity is correlated with **creativity**, intelligence, improved **learning** and memory, and problem **solving**."

BRENE BROWN

@MACKFOGELSON
THE READY

I often get questions about how to teach curiosity to a team.

Model it.

Try — for just one week — not to answer any questions. Every time someone on your team asks you a question in search of an answer from you, ask them one in return. Explore removing yourself from the role of ‘I have all the answers’ and watch what happens.

And in this process, also pay attention to what’s happening in your system that would squash the behavior of curiosity. If you’re modeling curiosity as a leader but there’s retaliation anywhere in the path along this process, you’ll kill any hope of building this muscle in your people.

Model curiosity and be curious about what needs to change in your system.

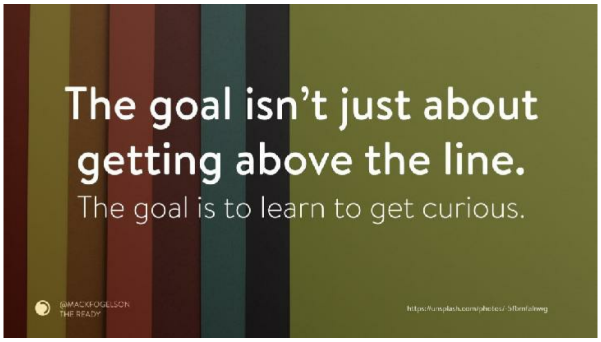
Additionally, a really powerful curiosity and self-awareness tool you can practice with your team is locating yourself. I’ve shared this video before, but even if you’re familiar with it, it’s worth watching again.

---

YouTube Video:
Locating Yourself - A Key to Conscious Leadership

---

What comes up for you when you watch that video? Where do you typically find yourself during your week at work? How about at home? The above and below the line language will help your team name their ways of being, especially when they aren’t showing up as their best selves. That’s OK. The point is to be curious about it.



The goal isn’t just about getting above the line.

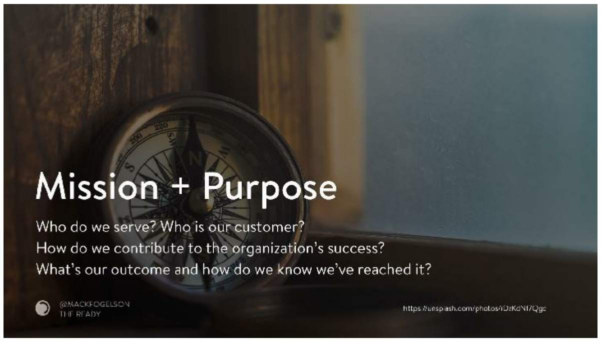
The goal is to learn to get curious.

@MACKFOGELSON
THE READY

https://unsplash.com/photos/-5fbmfalnwg

Teaching curiosity to your teams is teaching them to be vulnerable and courageous. To examine how they are showing up and learning to make the shift from blaming others to first getting curious about what’s really there for them. Learning to be curious is an opportunity to be brave enough to face fear, discomfort, and uncertainty.

As you’re working through surfacing things your team needs to say and hear, and practicing new ways of being, your team will be ready for a charter. Going through this exercise to establish a team charter and an operating rhythm is an important step to owning their ways of working.



Mission + Purpose

Who do we serve? Who is our customer?
How do we contribute to the organization’s success?
What’s our outcome and how do we know we’ve reached it?

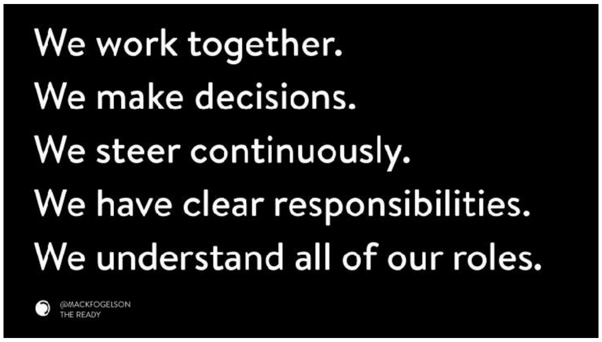
@MACKFOGELSON
THE READY

https://unsplash.com/photos/iDzKdNI7Qgc

When working with teams to define their mission and purpose, one of the questions we often ask is, ‘what will be different about the world after we’re gone?’

That may be a great question for an executive to answer, but at the team level — especially when the team is not used to being given the freedom to identify what purpose looks like for them — it can be pretty intimidating.

Identifying mission and purpose at the team level can start by understanding what the team's work actually is. What have we come together to accomplish and why? What if our team wasn't here? What would be lacking from the organization? Who is our customer and what is the value we provide them?



We work together.
We make decisions.
We steer continuously.
We have clear responsibilities.
We understand all of our roles.

@MACKFOGELSON
THE READY

What does it look like for your team to live its mission and purpose? You may:

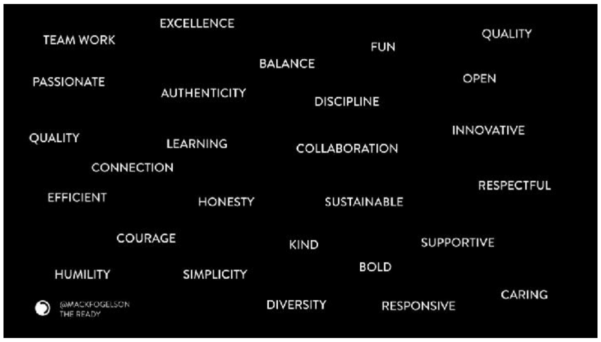
1. Name all of the roles required to do your work and deliver value to your customer
2. Clarify the specific responsibilities of each role and decouple these roles from a specific individual (each person may hold more than one role)
3. Choose to work towards outcomes, identifying the work required to achieve them each week
4. Actively use decision processes — like an advice process or integrative decision making — so we are constantly making small moves, learning, and moving forward
5. Invest the time — even outside of work — to be a team

Values + Guiding Principles

How do we want to show up every day?
How do we want to get your work done?
What will remind us to stay true to ourselves, our values, and our purpose?

https://unsplash.com/photos/A-NVHPka9Rk

Clarifying what you value as a team and what commitments you're making to each other is really important to steering towards your purpose.



EXCELLENCE
TEAM WORK
QUALITY
FUN
BALANCE
PASSIONATE
OPEN
AUTHENTICITY
DISCIPLINE
INNOVATIVE
QUALITY
LEARNING
COLLABORATION
CONNECTION
RESPECTFUL
EFFICIENT
HONESTY
SUSTAINABLE
COURAGE
SUPPORTIVE
KIND
BOLD
HUMILITY
SIMPLICITY
CARING
DIVERSITY
RESPONSIVE

@MACKFOGELSON
THE READY

One of the ways you can help your team define its values looks like this:

1. **Identify your top 5 personal values**Have each person on your team choose their top 5 values

as an individual. Each person takes the time to write it first on a sticky note — ranked from most important to least important. Then go around the room and allow each person to share as you scribe them for everyone to see.

2. **Identify your top 5 team values**
   Now that everyone is clear on individual values, have each person identify their top 5 values as a team. What is important to you about working with each other? Each person takes the time to write it first on a sticky note and then can share with the team.
3. **Vote for values**Using dot voting, have the team each place a dot next to each value they would like to make the list. Then, collectively, choose your top 5.
4. **Turn values into guiding principles**
   Break into pairs of 2 or groups of 3 and give each group the task of creating a statement about each value. If you were to turn that value into a line of guidance for your team, what would it say? Come back together as a group to share and finalize guiding principles.

Meetings

What is the purpose of each of our meetings?
How often and how long do we meet?
What meetings are missing and which ones do we need to stop having?

@MACKFOGELSON
THE READY

https://unsplash.com/photos/26MJGnCM0Wc

And as you may have learned with purpose and values, it’s not just about what it sounds like or getting it stenciled on the walls or captured in your employee handbook. Don’t just speak these things. Live them.

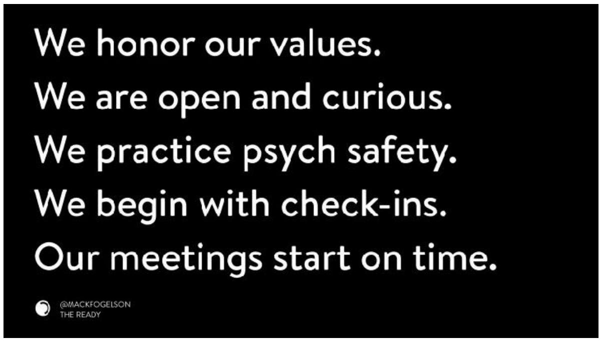
One of the most powerful things I’ve seen with living values is an executive team who chose to work their values into one of their operating meetings. They dedicated the first 10 minutes to one of their values turned guiding principle. They had been committed to listening more than they talk, so every time they checked in at that meeting, they would ask: What is the ratio for how much you’ve talked vs. listened since we last saw each other?

This executive team would work through this guiding principle for several weeks until it became a common practice in their ways of being. Once they felt it was a habit, they were ready to move on to the next one.

With any of this work, I am not advocating for adding anything to your existing operating rhythm. Rather, I’m challenging you to assess what is currently not serving you and even take something away. Especially when it comes to meetings.

Most of us spend the majority of our hours at work in meetings. Then, once we’ve finished our day, we spend time doing our work at home. What if it didn’t have to be this way?

Get curious about your meetings. Take a minute to write down all of the meetings you have with your team each week. What’s the purpose of each meeting? How often and how long do you need to meet? Are there meetings that are missing that would help us do our work (like a design or collaboration session every other week)? Which meetings can be repurposed or even eliminated all together? Maybe get crazy and stop having meetings for a week and notice what falls apart (or doesn’t)?

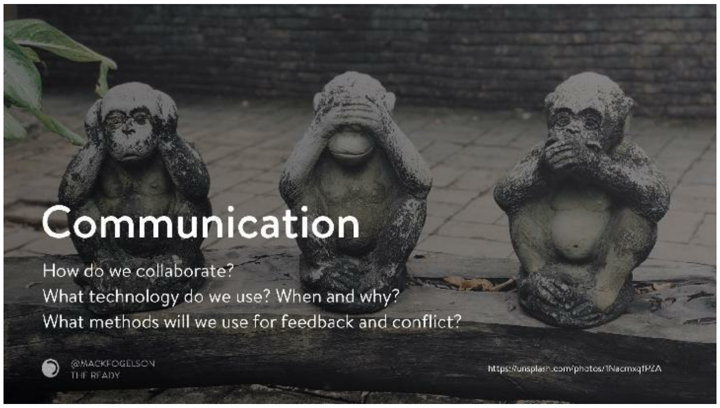


We honor our values.
We are open and curious.
We practice psych safety.
We begin with check-ins.
Our meetings start on time.

@MACKFOGELSON
THE READY

What might your meetings look like if your team is practicing your ways of being and living your purpose? You may:

1. Start meetings on time
2. Begin with check-ins (and end with a “what did you notice?” or “how did you show up?”)
3. Commit to practicing psychological safety
4. Stay open and curious and when we’re not, naming when we are below the line
5. Honor and work through the values you’ve committed to



Communication

How do we collaborate?
What technology do we use? When and why?
What methods will we use for feedback and conflict?

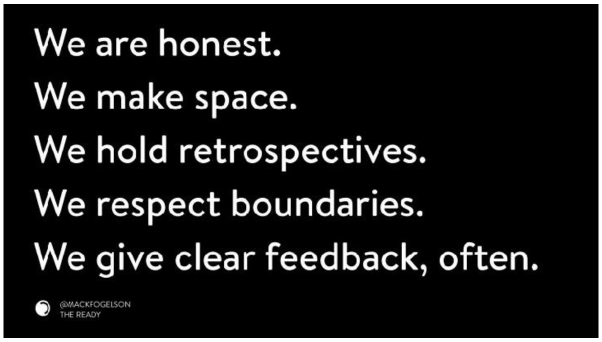
@MACKFOGELSON
THE READY

https://unsplash.com/photos/1NacmxqfPZA

Next, apply some curiosity to the way your team communicates. Do you collaborate with each other? How does that work? If there are silos, why is that? What technology do you need to do your work? How will you communicate when you need something? When we have feedback for each other, how do we deliver that? When we need to conflict about something, what’s the best way to do that?

This is also where some of your boundaries may come in. In Dare to Lead, Brene Brown offers some guidance about conflict and reminds us that our job is not to take responsibility for other people’s emotions. When you’re having conflict with someone, you can be angry, frustrated, or even surprised and sad, but make sure that behaviors stay above the line. You can communicate boundaries during conflict by simply naming guidelines like:

- I know this is a tough conversation. Being angry is okay. Yelling is not okay.
- I know we’re tired and stressed. This has been a long meeting. Being frustrated is okay. Interrupting people and rolling your eyes is not okay.
- I appreciate the passion around these different opinions and ideas. The emotion is okay. Passive-aggressive comments and put-downs are not okay.

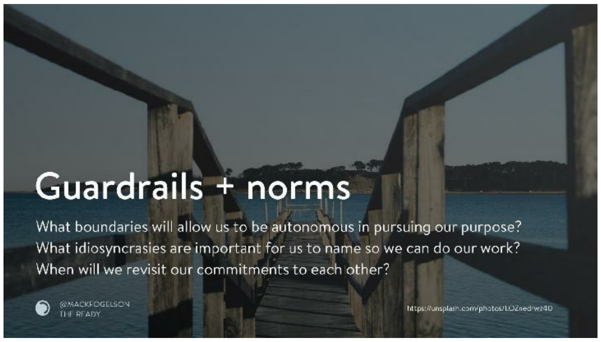


We are honest.
We make space.
We hold retrospectives.
We respect boundaries.
We give clear feedback, often.

@MACKFOGELSON
THE READY

Communication is such an integral component to how your team functions. If your team is communicating like champs, you may be:

1. Giving feedback that is clear and direct when it is warranted
2. Respecting the boundaries you have named for how you will work together
3. Preserving time to hold retrospectives so that you are learning from how you're working together, building trust and psychological safety
4. Making space to process conflict with each other and have hard conversations in front of each other
5. Risking vulnerability and having the courage to be honest with ourselves and each other



Guardrails + norms

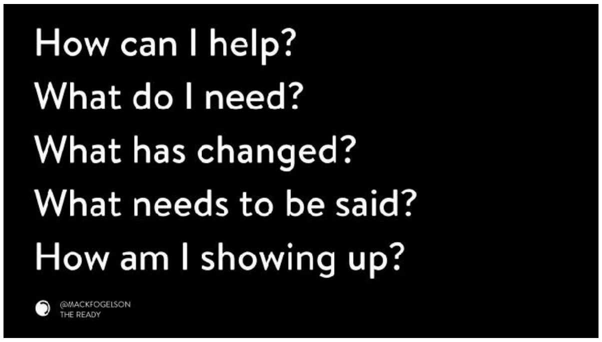
What boundaries will allow us to be autonomous in pursuing our purpose?
What idiosyncrasies are important for us to name so we can do our work?
When will we revisit our commitments to each other?

@MACKFOGELSON
THE READY

https://unsplash.com/photos/LOZnedrwz40

Guardrails and norms play a really important role in your team's chartering because naming them are a big step towards allowing your team to be autonomous in achieving its purpose.

Some questions that may guide your conversation with guardrails and norms: What are our decision rights? How much money can be spent before anyone needs to be asked for permission? Do we trust each other to work from home? When do we want to have a conversation about these commitments we're making to each other?



How can I help?
What do I need?
What has changed?
What needs to be said?
How am I showing up?

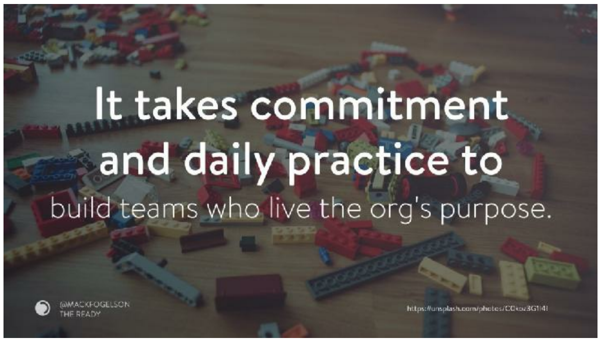
@MACKFOGELSON
THE READY

Living your guardrails and norms may require getting curious and asking questions at random intervals (maybe in your retrospectives) like these:

1. How have I been showing up with my team and co-workers?

2. Is there something that needs to be said that I'm not saying?
3. Has something changed in our team that has caused tension?
4. What does support look like right now?
5. How can I help someone else on my team?

Once all of this good work on your team charter is done, make sure it's available in a space that is accessible by your entire team and can be updated as you choose.



It takes commitment and daily practice to build teams who live the org's purpose.

@MACKFOGELSON
THE READY

https://unsplash.com/photos/C0koz3G1I4I

Building purposeful teams requires daily investment and practice from every individual involved. As with any methodology or practice, team chartering is not a silver bullet and is certainly not a short process. Depending on the level of dysfunction, it could take 3 to 4 months for your team to be ready and willing to make the space and work through their fears and concerns.

Although you may feel pressure to check the components of your charter off the list, practice patience, letting it be, and allow it to unfold over time. Your team — and organization — will be better for it.

PRACTICE
WHAT YOU
WANT

@MACKFOGELSON
THE READY

In the end, it's not just about achieving purpose and crafting a better workplace, it's about being better humans. Choosing to change your ways of working across an organization — both doing and being — is no small feat. But, big things can happen by starting with just one small purposeful team.

***

To read this article online with embedded hypertext links, go here:

**https://medium.com/the-ready/the-truth-about-team-chartering-21a7d5ead111**

# About Mackenize Fogelson

For sixteen years, **Mack** Fogelson has been guiding companies — and humans — into better versions of themselves. She is a powerful, courageous, wholehearted change-maker experienced in large-scale cultural transformation and systems change with Fortune 500 companies. Mack coaches executives, their teams, and organizations to lasting behavioral, systems, and cultural change so they can become more adaptive, resilient, and human. She has been honored to move thought forward with publications such as The Economist, is an accomplished writer, speaker, former entrepreneur, and Certified Dare to Lead™ Facilitator. Mack lives in Fort Collins, Colorado with her husband Jon and her mini-Macks: Ryan and Easton.

# Proper Scaling of Scrum and Dynamic Financial Forecasting

By Gene Gendel

Posted on February 5, 2018

The purpose of this post is to summarize two very important and independent topics and then integrate them together, into a joint discussion.  The topics are:

1. *Moving from rigid annual budgets to rolling forecasts (super important! in agile/adaptive product development environments)*
2. *Quality of scaling in agile product development, specifically Scrum*

   *…and **tying effective scaling of Scrum to dynamic financial forecasting**.*

---

## Rigid Annual Budgets vs. Dynamic/Rolling-Wave Forecasting

Challenges presented by rigid annual budgets have been known for a long time.  For people that are new to the topic, a great way to stay on top of most recent research and publications, is to follow what is going at BBRT.org (Beyond Budgeting Round Table).  One of BBRT's core team members – Bjarte Bogsnes, in his book "*Implementing Beyond Budgeting: Unlocking the Performance Potential*" (please, refer to the book's highlights here),  clearly summarizes the problems with conventional, end-of-year rigid budgets. They are as follows:

1. Budgets represent a retrospective look at past situation and conditions that may not be applicable in a future
2. Assumptions made as a part of a budgeting process, even if somewhere accurate at the beginning, get quickly outdated
3. Budgeting, in general, is very time-consuming process, and it adds additional, financial overhead to organizations
4. **Rigid budgets, can prevent important, value adding-activities, and often lead to fear of experimenting, researching and innovating (crucial for incremental development)**
5. Budget reports are frequently based on subjective metrics, as they take on the form of RAG statuses, with the latter, introducing additional errors and omissions (for details, please refer to Red, Yellow, Green or RYG/RAG Reports: How They Hide the Truth, by M. Levison and The Fallacy of Red, Amber, Green Reporting, by G. Gendel)
6. Budgets, when used as a yardstick to assess individual performance, often lead to unethical behaviors (e.g. "churning & burning cash"at year-end to get as much or more next year) or other system-gaming activities

…The list of adverse effects caused by traditional budgeting is long…

On contrary, a **rolling-wave forecast**, respects the fact that environmental conditions are almost never static, and recognizes that if too much reliance is placed on prior years' financial situation, it may lead to miscalculations. Rolling-wave forecasts are based on frequent reassessment of a small handful of strong KPIs, as oppose to large number of weak KPIs, as frequently done in conventional budgeting.  The more frequently forecasts are being made, the higher chance that most relevant/reliable information will be used in assessments.  One good way to decide on cadence of rolling forecasts is to align them with meaningful business-driven events (e.g. merchandise shipments, production code deployments, etc.).  It is natural to assume that for incremental/iterative product development (e.g. Scrum), when production deployments are made frequently and in small batches, rolling-wave forecasting could be a concurrent financial process.  Short cycle time of market feedback could provide good guidance to future funding decisions.

It is worth noting that one of the key challenges that Scrum teams face today, is the “iron triangle” of conventional project management, with all three of its corners (time, scope, budget) being rigidly locked. And while the most common approach in Scrum is to make scope flexible, ‘clipping’ the budget corner brings additional advantage to teams. Above all other benefits, rolling-wave forecasts address the problem described in **#4 above**, as they provide safety to those teams that want to innovate and experiment.

But what if there not one but many Scrum teams, each working on their own initiatives, running under different cadences (asynchronized sprints) and servicing different customers? How many independent rolling-wave forecasts can one organization or department adopt before things become too complicated? What is too much and where to draw a line?

Before we try to answer this question, let’s review what is frequently seen, when organizations attempt to scale scrum.

## Proper Scaling vs. “Copy-Paste” Scaling

Let’s look at the following two situations: (**1**) more than one Scrum team, independently, doing *their own* Scrum and (**2**) more than one Scrum team, working synchronously, on the same product, for the same customer, sharing the same product backlog and domain knowledge. The former case, is referred to as **“Copy-Paste” Scrum**, clearly described by Cesario Ramos. The latter case can be seen in skillful **Large Scale Scrum (LeSS)** adoptions. Here are some of the most classic characteristics of both scaling approaches:

**(1) – “Copy-Paste” Scrum**

- Product definition is weak. Applications and components that don’t have strong customer alignment are treated as products
- “Doing Scrum” efforts are often a result of trying to meet goals of agile transformation (some annual % goals must be met), set at enterprise level
- Tight subsystem code ownership
- Top-down, “command & control” governance, with little autonomy and self-management at team level
- Importance of Scrum dynamics and its roles are viewed as secondary to existing organizational structure blueprints
- Too many single-specialty experts and very few T-shaped workers
- No meaningful HR changes to support Scrum team design

**(2) – Large Scale Scrum (LeSS)**

- Simplified organizational design. Reduction of: silos, handovers, translation layers and bureaucracy
- Scrum is implemented by coordinated, feature-centric teams, working on widely defined Product, for the same PO.
- Local Optimization by single specialists is eradicated
- Scrum is a building block of IT organizational structure
- Teams are collocated Multi-site development is used for multiple locations
- Strong reliance of technical Mentoring and Communities of Practice
- No subsystem code ownership
- Reduction of “undone” work and “undone department”
- Focus on Customer values
- Strong support by Senior Leadership & intimate involvement of HR

**Note**: *Please refer to Scaling Organizational Adaptiveness (a.k.a. “Agility”) with Large Scale Scrum (LeSS), for additional graphic illustration.*

Based on the above, the following also becomes apparent:

In “copy-paste” Scrum, development efforts, marketing strategies and sales (ROI) are not treated as constituents of the same unified ecosystem. In this scenario, it is almost impossible to fund teams by means of funding real, customer-centric products. Why? There are too many independent ad-hoc activities that take place and artifacts that are created. There is no uniformed understanding of work size and complexity that is shared by all teams. Estimation and forecasting made by each individual team is not understood by other teams. Team stability (and subsequently, cost-per-team member) is low, as individuals are moved around from project to project and shared across many projects. Further, with multiple teams reporting into different lines of management, there is a much higher chance of internal competition for budget. By the same token, there is a low chance that a real paying customer would be able to step in and influence funding decisions for any given team: too many independent and competing requests are going on at the same time.

In organizations, where "copy-paste" Scrum is seen (and is often, mistakenly taken for scaled scrum, due to lack of education and expert-leadership), there is still strong preference for fake programs and fake portfolio management. Under such conditions, unrelated activities and, subsequently, data/metrics (often fudged and RAG-ed) are collected from all over the organization and "stapled" together. All this information rolls up to senior leadership, customers and sponsors. Subsequently, what rolls down, is not dynamic funding of well-defined customer-centric, revenue-generating products, but rather rigid budgets for large portfolios and programs that are composed of loosely coupled working initiatives, performed by unrelated Scrum teams (secondary, to conventional departmental budgeting). As rigid budgets cascade down from top, onto individual teams, they further solidify the "iron triangle" of conventional project management and hinder teams' ability to do research, experimentation and adaptive planning.

On the other hand, in Large Scale Scrum, things are different:

- When up-to-eight LeSS teams work synchronously, together (side-by-side), on the same widely-defined product (real), their shared understanding of work type and complexity (having certain scrum events together really helps!) is significantly better. As a result, when it comes to forecasting a completion of certain work (features), eight LeSS teams will do a better job than eight loosely coupled teams that work completely independently, on unrelated initiatives.
- Since all LeSS teams work for the same customer (Product Owner), there is a much higher chance that they will develop a shared understanding of product vision and strategy, since they are getting it from an authentic source – and therefore will be able to do planning more effectively.
- Having more direct correlation between development efforts LeSS teams (output, in the form of shared PSPI) and business impact (outcome, in the form of overall ROI), makes strategic decisions about funding much more thoughtful. When real customers can directly sponsor product-centric development efforts, by getting real-time feedback from a market place and deciding on future strategy, they (customers) become much more interested in dynamic forecasting, as it allows them to invest into what makes most sense. Dynamic forecasting of LeSS, allows to increase/decrease number of scrum teams involved in product development flexibly, by responding to increased/decreased market demands and/or product expansion/contraction.

Noteworthy that in LeSS Huge cases, when product breadth has outgrown capacity of a single Product Owner and requires work by more than eight teams, dynamic forecasting can still be a great approach for Product (overall) Owner and Area Product owners (APO): they can strategize funding of different product areas and make necessary timely adjustments to each area size/grown, as market conditions change.

---

## Conclusion:

All of the above, as described in LeSS scenario, will decrease organizational dependency on fixed budgets, as there will be less interest in outdated financial information, in favor of flexibility, provided by rolling-wave forecasting that brings much closer together "the concept" (where value is built – teams) and "cash" (where, value is consumed – customers).

***

To read this article online with embedded hypertext links, go here:

**http://www.keystepstosuccess.com/2018/02/proper-scaling-of-scrum-and-dynamic-financial-forecasting/**

# Centralized vs. Decentralized Coaching

By Gene Gendel

Posted on May 20, 2018

### Key Takeaways

- There is a frequently seen confusion with respect to the definition of agile coaching: coaching focus (e.g. enterprise vs. team) is confused with coaching alignment (centralized vs. decentralized) within an organization
- Centralized coaching departments run the risk of turning into a single-specialty organizational silos that are locally optimized for their own expansion and personal success; they are also removed from real action. The reasoning behind: standardization - has its weaknesses.
- Centralized coaching is often limited to being "responsible for introducing KPIs, documentation of script-style-one-size-fits-all best practices and cookie-cutting approaches". This leads to system gaming by other departments and organizational silos that must "meet numbers goals"
- Centralized Agile coaching makes sense only when it takes place within an organization that is small enough to be effectively managed front-to-back (including its all organizational layers) and is genuinely supportive of its own coaches, by providing them with "organizational immunity" and operational safety - to enable them perform their challenging duties
- The main advantage of decentralized coaching approach is that coaches are close to real action: deeply engaged with products/services, and are intimately engaged with senior leadership. Decentralized coaching is deep & narrow (as opposed to being broad and shallow) and takes time to cause meaningful and sustainable organizational changes.

  *The old rules no longer apply...*

  *When General Stanley McChrystal took command of the Joint Special Operations Task Force in 2004, he quickly realized that conventional military tactics were failing. Al Qaeda in Iraq was a decentralized network that could move quickly, strike ruthlessly, then seemingly vanish into the local population. The allied forces had a huge advantage in numbers, equipment, and training—but none of that seemed to matter….*

  *A new approach for a new world...*

  *McChrystal and his colleagues discarded a century of conventional wisdom and remade the Task Force, in the midst of a grueling war, into something new: a network that combined extremely transparent communication with decentralized decision-making authority. The walls between silos were torn down. Leaders looked at the best practices of the smallest units and found ways to extend them to thousands of people on three continents, using technology to establish a oneness that would have been impossible even a decade earlier. The Task Force became a "team of teams" - faster, flatter, more flexible - and beat back Al Qaeda."*

  ***Original source**: Amazon summary of the book "Team of Teams", by General Stanley McChrystal (Author), Tantum Collins (Author), David Silverman (Author), Chris Fussell (Author)*

**Note**: This writing was inspired by the discussion among LeSS trainers (CLT) & candidates, LeSS-Friendly Scrum Trainers (LFST), Certified Enterprise Coaches (CEC) and Certified Scrum Trainers (CST). The main influencers of this writing are: Rowan Bunning, Josef Scherer, Greg Hutchings, Michael Mai, Robin Dymond, Viktor Grgic, Bas Vodde and Gene Gendel. Points of view are drawn on experience of individuals' consulting at various companies, in the following industry sectors: Global Telecommunications, Finance/Banking, Insurance, and Department of Defence.

## Resolving the Confusion: Focus vs. Position

While this writing is about the differences in effectiveness of Centralized Coaching vs. Decentralized Coaching in complex organizational settings, first, we need to clear up one important misunderstanding: mistakenly, enterprise-level coaching is confused with centralized, while team-level coaching is confused with decentralized.

This is a confusion of two different coaching aspects: focus and position. For example, within an internal organizational structure of the same CTO (could be different from other CTOs of the same organization), there could be both levels of coaching: enterprise- and team- level coaches - would be working concurrently, while complimenting each other's work in many ways. What makes them different is their coaching focus (enterprise dynamics vs. team dynamics). But at the same time, their placement/sense of belonging is the same: they are decentralized from an enterprise apex and fit into a more local area (sphere of influence of one CTO).

To summarize the definitions of two coaching aspects:

**Coaching focus**:

- **Team** coaches are primarily focused on tools, frameworks and dynamics of multiple teams, with less emphasis on organizational transformation.
- **Enterprise (organizational)** coaches - are more focused on organizational dynamics and more abstracted elements of transformation, with emphasis on senior leadership, upper management, organizational policies (e.g. HR), and multiple organizational domains.

Both focus areas, enterprise - and team-level - are equally important and required for transformational success, irrespectively of where a coaching discipline is placed: centrally or de-centrally. Noteworthy, many experienced coaches are able to operate equally effectively at team- as well as enterprise level, as they 'travel' up and down an organizational vertical. (Note: for more detailed definitions of coaching focus areas, please refer to authentic and credible sources describing this profession: Scrum Alliance and LeSS.works.)

**Coaching position**:

- **Centralized** - a distinct organizational unit (e.g. Agile Center of Excellence or Agile Global Centre) that drives agile transformation across an entire organization, by introducing best practices, tools, techniques, standards, benchmarks and scorecards, against which everyone else is measured. Such organizational unit is loosely coupled with any specific product, service or line of business. It is primarily supported (and sponsored) by an organizational structure that has been selected and 'put in charge' by more senior leadership. With this selection, other organizational structures (e.g. operations or product groups) usually remain less vested in the effort, and even if follow along, do so with noticeable complacency.

- **Decentralized** - is a less rigidly structured team of like-minded coaches that align themselves with a clearly defined product, service or line of business. Transformational focus here is much narrower, and requires much more jenuine support/vesting ( with sponsorship!) from multiple organizational verticals involved (e.g. business, operations, IT, HR, finance, etc.). In order for decentralized coaching to have a meaningful organizational impact, an organization must be of manageable size (no Big Bangs), as it is defined by an organizational sushi roll that contains elements/instances of multiple organizational structures involved.

Let's take a closer look at two distinct types of a coaching position:

## Centralized coaching

Usually, this approach is preferred by organizations that treat agile transformation, as a trend, accompanied by inspirational slogans, PR and town hall talks, with senior organizational leaders rendering support mainly "in blessing and in spirit", rather than by real actions. This is frequently done without real understanding of deep system implications of this important undertaking. There is no true gemba by senior leadership. Actual transformational efforts are delegated downward-and-downward, to lower echelons of power, where the original purpose is diluted and focus is lost. With agile coaching, being a centralized organizational function that owns transformation, one of its main deliverables becomes setting of standards and measures of success, by which the rest of an organization is measured.

A usual justification of centralized coaching function, is the need to standardize and define “best practices” for others, by claiming that “...too many independent adoptions would be hard to measure/compare against another…so we need consistency...” - is something that is very important for organizations, where individuals’ rewards and compensation are based on individual performance, scorecards and KPIs. To fulfill organizational mandates for measurements, centralized coaches are tasked with introducing agile maturity metrics (AMMs) that are usually composed of a wide array of maturity indicators, bundled together in some arbitrarily created maturity buckets/levels. **This approach very quickly turns into a box-checking exercise for other organizational units, whereas everyone tries to claim higher maturity, in order to meet goals. Not surprisingly, this is accompanied by system gaming and unsubstantiated claims of success.**

Since with centralized coaching approach there is a higher volume of demand for coaches and it often exceeds a supply, quality is frequently compromised, and it manifests itself as follows:

- Larman’s Law of Organizational Behaviour # 4 kicks in - it describes individuals, whose past roles have become less needed in flattened/leaner organizations, and now “...coaching seems like another thing these misplaced folks can do successfully...” These individuals perceive coaching as an opportunity to stay busy and fast-track their own careers (agile coaching for them is just a “hop-on-hop-off” bandwagon) until they secure another comfortable position in an organizational structure..

- Coaches “Centaurs” - describes low quality external consultants that are hired temporarily, from outside, at low/wholesale cost, through third-party vendors, conveniently listed in preferred vendor applications of client-companies. (Sidenote: usually, these vendors have challenges supplying high-quality intellectual assets in general, let alone agile transformation consultants, as the latter are relatively specific professional niche).

Since centralized coaches are responsible for setting the tone for the rest of organization, they are also tasked with producing large volumes of supportive documentation: standardized training materials, audios, videos, etc. Hundreds (at times, thousands) of internal wiki pages are created to host information that, for the most part, **comes in the form of copy-pasting what is already available publicly on internet (easily accessible commodity)**. This consumes many man-hours and produces a false sense of information ownership, internally. Such information also becomes outdated frequently, and requires lots of internal manual rework to be kept up-to-date.

As demand for coaching comes from various organizational areas, centralized coaches get temporarily deployed to offer assistance. But since for many internal customers, agile transformation is still a meeting-numbers game (to comply with enterprise-wide organizational mandates), demand for coaches often exceeds its supply (it comes in spikes). As a result, often coaches are spread thin across multiple organizational areas and their ability to truly make long lasting, meaningful impact is hindered. Coaching becomes broad and shallow. In a long run, as a delayed result of demand-spiking, there is an accelerated growth of centralized coaching group, as described above.

**… And now, artificially inflated group of centralized coaches, takes on the form of a single-functional specialty department that is governed by ‘local optimization’: they are optimized to maintain their own increased size and the need to stay busy.**

Here are some quotes about centralized coaching from the influencers of this writing:

From Viktor Grgic of Odd-e:

> *Organizational tree of agile coaches who commonly force upon others their “services” is quite serious dysfunction. If organization is very much into this, one might choose to limit scope of adoption, show real result while others are extremely busy with programs, etc. In other words, there is not much that can be done when KPIs for agile transformation are set at the very high level of organization and everyone is busy complying with them.*

From Greg Hutchings, of Amelior Services:

> *I would discourage those who think that the best use of agile coaching and training budgets would be to create an agile center with people primarily aligned with and focused on belonging to and spending time with a separate, specialist group, as this is just about the exact opposite of go-see, working at gemba, and inspecting and*

*adapting with people in the main value creation part of the organisation.*

### Leaving Your Men Behind

One of the most painful examples of centralized coaching dysfunction is having coaches internalized/commoditized by an organizational structure that does not genuinely support agile transformation, because it neither understands it, nor sees any personal benefit of it.  Furthermore, there is a fear that organizational agility at-large (at scale) might be viewed as a threat to the very purpose and usefulness of an organizational structure itself. For example, placing a coaching function within an existing Business Analysis group, or an existing Governance CoE, or management CoP/PMO or Architecture department would be a disservice to an agile coaching initiative altogether, as these organizational verticals would not provide coaches with necessary support and safely, to perform challenging duties. They will leave their coaches behind.  For example, if coaches reveal organizational dysfunctions that may lead to an increase of political tensions, unsupportive organizational structures and Taylorian managers of modern days will readily sacrifice their own coaches ("throw them under a bus") to regain political recognition to better fit an organizational landscape.

## De-Centralized coaching

With decentralized coaching approach, coaches are locally aligned/dedicated with teams, their customers & products, and immediately involved senior leadership.

This approach is usually preferred, when a specific organizational area (e.g. IT, product development) makes a **conscious decision** to improve its agility/adaptiveness.  Decentralized coaching is typically sponsored/supported by a real end-consumer, with enough organizational power to protect autonomy and authenticity of original transformation goals (e.g. CTO/CIO and respective senior business partners).  In this scenario, people that consume services and people that pay for services - **are the same people that are really vested in success.**

With decentralized coaching, fewer but more experienced and dedicated coaches are required.  Coaches are more carefully selected by an organization. Coaching seasoning/experience is being viewed as the most important factor and becomes a natural remedy against a 'wholesale' approach: highly qualified coaches will not work for a discounted pay, while genuinely vested clients are willing to pay a fair price for high quality service.

Decentralized coaching is deep and narrow: it is focused on fewer people (in total) but on a wider gamut of organizational elements/domains (e.g. IT, business partners, HR, finance) - by taking a holistic look at the whole organization.  With this approach, it is much easier to trace effectiveness of coaching "from concept to cash", by chasing not just superficial indicators/outputs (e.g. adherence to Scrum events, increased velocity, stable/collocated teams) but also seek true business outcomes (increased ROI, improved customer satisfaction, beating external competition, team happiness).

Once engaged deeply, dedicated coaches go through a few important steps of a coaching cycle: assessing, delivering structured training, coaching and gradually disengaging, while giving autonomy back to a client: there is no rush to meet year-end target numbers with coaching.  Dedicated (local) coaches and organizations they support, are much less preoccupied with KPIs, metrics, scorecards and meeting numbers. AMMs are treated solely as a barometer of local improvements (please see how).

Throughout an entire engagement period, coaches remain deeply embedded with development teams and respective product teams.  This is all accompanied by many observations, short feedback loops and frequent retrospectives. Any issues or observations that have systemic implications and require attention of senior leadership, are addressed together with senior leadership.

Given autonomy and sovereignty of an organization and its dedicated coaches, there is a higher chance of running experiments, inspecting & adapting, without fear of failure, or being prematurely judged and becoming a subject to repercussions.

## Why False Dichotomy?

Sometimes, we hear a concern that because of decentralized coaching approach, there will be no adequate shared learning across the whole organization. But why should this be the case? Why should decentralized coaching and shared learning be mutually exclusive (**false dichotomy**)? Could not be there some other effective ways to ensure that coaches succeed in both: remain dedicated to their own, distinct organizational areas, for the reasons described above, and are still able to collaborate, synergize, learn from one another and create full transparency for their individual methods and styles?

All of this could be very effectively achieved by forming self-organized/self-governed coaching communities of practice, where coaches from different organizational areas, of different focus (team-, enterprise-) and with different skill set (technical-, career-, process-) consistently share their knowledge and experience, in a safe, reporting-free environment.

## Conclusion

If an organization is relatively small and centralized coaching does not turn into a PR showcase, with a small group of privileged individuals, trying to set a tone for thousands of others, by enforcing KPIs, metrics and best practices, **AndIf** there is a way to prevent centralized coaches from rushing towards year-end target numbers, **AND** instead, engage deeply and narrowly with clients, while offering continuous support and conducting safe experiments, **THEN** centralized coaching approach is worth a try. **EndIf**

However, if the above conditions are not possible, because of historic organizational malfunctions, then dedicated coaches that are deeply embedded with vested customers, is a better choice.

Here are some more quotes from this writing's influencers:

From Bas Vodde, of Odd-e and the co-founder of Large Scale Scrum framework:

> *If you have a centralized team that can *truly* go to products and coach them, they're very valuable as they see a lot of cross-product dynamics. If they can't do that, then a decentralized group is a better chance of getting at least some value out of the coaching.*

From Rowan Bunning, of Scrum WithStyle:

> *If the goal is for Agile thinking and practices to be disseminated throughout an organisation in a way that everyone feels that they "own" their ways of working, then coaches should be co-located and deeply embedded with teams and business units they support. Furthermore, if a message from coaches to developers is to move from single function groups to cross-functional teams, then having coaches centralised into a single-function group may seem hypocritical. It may also be perceived as a pursuit of control over how Agile coaching services are procured and disseminated - potentially to the benefit of those in a centralised group.*

**Author's note**: By way of illustration, in LeSS adoptions, a LeSS coach would focus on a 2-8 Teams (about 50 people) that work on the same product, for the same Product Owner, out of the same product backlog. LeSS coach would also focus on additional organizational layers that are in immediate proximity to IT-side of LeSS organizational construct, specifically, on a product/product-people side, and their respective senior leadership. LeSS coaching engagement is meant to be deep and narrow (as oppose to broad and shallow), and is focused on a small "sushi roll slice" of a whole organization. For LeSS coaching to be successful, bigger does not mean better, and it naturally supports the idea of organizational descaling.

***

To read this article online with embedded hypertext links, go here:

**https://www.infoq.com/articles/centralized-decentralized-coaching/**

# About Gene Gendel

**Gene** Gendel is Agile Coach, Trainer and Organizational Design Agent. Gene is a proud member of the small community of Scrum Alliance Certified Enterprise Coaches (CEC). Today, he is the only CEC who resides in NY State. Gene’s goal is to help organizations and individual teams with improving internal dynamics, organizational structure and overall efficiency. He strives to engage at all organizational levels: senior- and mid-level management, teams and individuals. In his work, Gene uses various methods, tools and techniques to strengthen learning of others and to ensure that teams and individuals gain autonomy after he “coaches himself out of the job”. Throughout his long career, Gene has served small, mid-size and large companies, domestically and abroad. Gene is a well-recognized member of global and local agile communities, where he influences people via open-space agile collaboration workshops, coaching retreats, group events and presentations.

Gene is a well-recognized blogger and publisher. He is the co-author of the book *Agile Coaching: Wisdom from Practitioners* (free pdf). His collection of personal essays (“*The Green Book*“) can be also found here.
Gene strongly supports Scrum Alliance (SA) in its efforts of *“transforming the world of work”*. He is an active member of SA working group of coaches and trainers that have been involved in improving SA certification/education programs, by aligning them with natural career paths of agile professionals: Team Level Coaching Certifications (CTC) (Gene is also one of co-creators of the program) and Enterprise Level Coaching Certifications (CEC). Through these efforts, Gene tries helping highly qualified individuals to differentiate themselves from lowered standards.

Gene’s additional credentials are:

Certified Team Coach (CTC)
Certified in Agile Leadership (CAL)
Certified in Large Scale Scrum (CLP)
LeSS-friendly Scrum Trainer(LFST) , who is able to teach LeSS Basics Certification (CLB) course, according to CLB Learning Objectives, and Certified LeSS Trainer (CLT) -candidate
Certified in Scrum @ Scale (S@S)
Mentor to Coaches
CSM, CSPO, CSP, PMP

# 10 Components that Successfully Abolished Hierarchy (in 70+ companies)

By Lisa Gill

Posted on March 22, 2018

1
TRANSPARENCY
2
No Hierarchy
3
self managing teams.
no priveleges
4
5
salary balance
6
no controls
7
measure & track everything
8
shared decision making
9
no layoffs
10
profit share
10 ESSENTIALS OF
SELF MANAGEMENT
TRANSFORMATION

Sketch by Andy de Vale

I recently flew to Lisbon to participate in a two-day talk and workshop led by Pablo Aretxabala and Jabi Salcedo from K2K Emocionado. K2K has successfully transformed 70 organisations (many of them industrial companies) in the Basque Country from traditional hierarchies to flourishing, self-managing organisations. That they have successfully transformed and improved results (profitability, productivity, absenteeism, salaries) in so many companies with such consistency piqued my interest. So how do they do it?

## People who believe in people

At the core of their approach is the former CEO's concept of "nuevo estilo de relaciones" or "**new relationship styles**." K2K supports companies — usually for a period of three years — to radically change their organisational structure, eliminating managers, controls, and privileges, and implementing transparency, shared responsibility, and profit-sharing. Before they even start the process, employees have to *choose* it. As Jabi says: **"Do nothing unless you have asked people first and involved them…"** (you can learn more about that in this previous Corporate Rebels post).

There are 10 essential components of the K2K method of transformation. The government approached them at one point and said, "You've transformed nearly 70 organisations in the Basque Country and we have 22,000. Could you do some more? Maybe if you took out some of the more radical steps, more companies would be interested?" The team went back and had meetings for several days to explore if this was possible. In the end, they decided that each of these 10 components was important — without all 10, you simply won't get the benefits. So here they are:

## 1. Total transparency

Trust is a key factor and so one of the first steps is to open up all information and create accessible financial reports for everyone. K2K helps the company set up monthly and quarterly meetings to inform and train people about how to understand the financial information.

## 2. No hierarchy

K2K spends two months interviewing people in the company and analysing its current form and what people want, after which they call a General Assembly and present the new, proposed structure. They eliminate the whole middle management layer, but never layoff anyone (more on that in a moment).

The CEO becomes General Coordinator and no longer has decision-making power over anyone. Instead, they now devote their time to caring for people and making their work easier. Jabi joked that one company they worked with had 70 people and nine levels of hierarchy and even more salary levels — and they wondered why people didn't work as a team!

## 3. Self-managing teams

Companies give K2K the authority to propose the team structure (rather than create it themselves) because they have lots of experience and expertise in doing this. Decision-making is established as a shared responsibility and every decision involves the people who are impacted. To streamline this, they create a Pilot Team which contains a representative of every team and has monthly meetings.

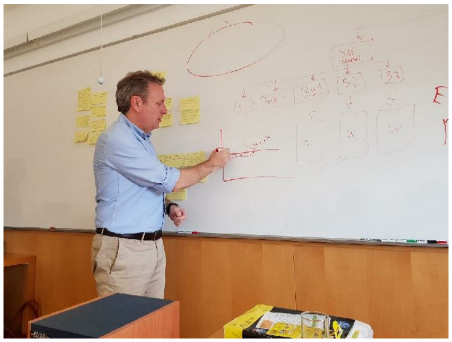
Teams also have Commitment Meetings every two to three weeks to discuss commitments against team objectives. Each team elects a 'leader' as a representative, however they have no power or extra salary. This person serves for only one or two years because after that, Jabi says, "people start to be a boss." Teams can share this role across two or more people, and replace the leader at any time.

## 4. No privileges

From day one, all privileges are eliminated: closed offices, executive dining rooms, reserved parking spaces, any bonuses or incentives for individuals' performance, and no special access to information.

## 5. A fair salary balance

When K2K reveals the new organisational structure to the team, they also share the new salary levels. Overtime pay is scrapped and the lowest salaries are lifted to minimise the gap between salaries. Nobody's salaries are lowered (for example the former middle managers), however one rule is that the highest earning 10% can earn no more than 2.3 times the salary of the lowest earning 10%. On occasion, this has resulted in an agreement that the CEO lower their salary.



Pablo explaining how they lift the lowest salaries and standardise salary levels

## 6. No controls

In factories, the clocking in and out system vanishes. "We move by goals and commitments, not by hours," explained Pablo. The team is in charge of resolving any instances of inappropriate behaviour when they occur. It's also essential that people are allowed to make mistakes. In the first six months, it is almost always total chaos and frustration. But what first-time manager hasn't made mistakes? Making mistakes is key to learning and must not be punished or discouraged. As Pablo put it: "**Sometimes you succeed, sometimes you learn.**"

## 7. Measure and track everything

The measuring and tracking here isn't to control, but so that teams have realtime information about how they are doing. The K2K approach also involves measuring teams only, and never individuals.

## 8. Shared decision-making

To begin with, K2K helps facilitate decision-making discussions as it's very new for teams to be making decisions without a boss. To know if they're making a good decision, K2K suggests people ask four questions:

1. Will it be better than before? (Think long-term)
2. Can we explain it to **everyone**? (Is it transparent or not?)
3. Has everyone who'll be affected by the decision and who has relevant, important knowledge been involved or consulted?
4. If we put ourselves in the shoes of those who'll be impacted, would we make the same decision?

Most decisions can be made in teams, however decisions that impact a larger number of people are agreed in General Assemblies.

## 9. No layoffs

Remarkably, K2K has never laid off a single person in their transformation processes. Middle managers are supported to find new roles. This contributes to offsetting the increased cost of increasing many of the salaries because, as Jabi explained, 15% of the workforce that was previously controlling others is now dedicated to improving efficiency. And the large percentage of the workforce that was being controlled is now highly effective because they've been set free to concentrate on the things that matter!

## 10. Sharing profits with all

In any K2K transformation, this is a non-negotiable: 30% of the profits must be shared with all employees. "Without this, everything else is just smoke," Pablo told us.

Jabi explaining the K2K transformation process

## So, where do you start?

If, like the participants in our workshop, you're excited and itching to start a K2K-style transformation of your own, here are a few suggested starting points from Pablo:

1. **What do you feel?** — many times Pablo told us the results and the 'how' are less important than how you feel. In fact, K2K's tagline is "Feel, think, do." Having read this piece, how do you feel? Inspired? Scared? Sceptical? Activated? Challenged? Get present to that.
2. **Get informed** — geek up on new relationship styles in organisations: read, watch videos, visit progressive companies, have conversations, contrast with your own situation.
3. **Really, really want it** — a transformation like this takes grit and commitment and will never work if you don't truly want it. If you're doing it for the results or because it's the latest trend, you can harm others and yourself.
4. **Share and decide** — share your vision with others and decide to change by involving everyone that will be part of the transformation. Pablo said it's important to make this decision formal. As humans, the power of declarations ("I now pronounce you man and wife") and rituals profoundly influences our commitment and personal transformation.
5. **Enjoy the road because there is no finish line** — there's no end-point to a transformation. Jabi said there isn't a single company that's working 100% as they'd like it to in their portfolio, just as there isn't a single traditional company out there in the world performing 100% as we'd like it to. To start this process is to embark on a lifelong journey of learning.

And finally, **find your fellow rebels.** If you embark on this journey as an organisation, you'll be in the minority and sometimes it will feel lonely and frustrating. Connect with other radical people and organisations for support. K2K founded Ner Group in 2009 for this reason, a network that organisations they have helped transform can join to offer each other support, knowledge, and even pool resources.

## What's next for K2K?

K2K continues to evolve. They're discovering that they could offer a lot more training to people in the companies they transform to help them adapt to this new way of working less painfully. They're also exploring bringing in psychologists and coaches to support the softer side of transformation and foster the skills needed for self-managing teams to thrive. Their story is certainly starting to garner attention and interest and I suspect they'll be expanding across Europe and beyond soon.

***

To read this article online with embedded hypertext links, go here:

**https://medium.com/the-ready/10-components-that-successfully-abolished-hierarchy-in-70-companies-b90265d019ad**

# About Lisa Gill

**Lisa** Gill is an author, coach, consultant and leadership trainer. She supports teams and organisations around the world to be decentralised, self-managing and participatory. Her focus is on the "human skills" needed for self-organisation to thrive. Lisa is also the host of the Leadermorphosis podcast and has interviewed some of the world's leading thinkers and practitioners about the future of work. She is the founder of Reimaginaire and a trainer at Tuff Leadership Training.

# How Courage Can Create Safety

By Ellen Grove

Posted on April 4, 2018

True story: During a brand new team's liftoff workshop, where they were figuring out their team's identity and purpose and working out how to work together, conversation revealed that several team members were very uncomfortable addressing disagreements and conflicts directly. Based on experiences with previous teams, the new colleagues realized this was a team dysfunction they needed to address if they wanted to succeed as a team. They decided to set a team safe-word (they chose "broccoli") that would be used to signal that this team needed to stop and address a problem directly rather than dodging it. Because they had agreed on this – and put it in their working agreement explicitly – even the conflict-averse team members felt comfortable later on "calling broccoli!" when touchy problems did arise.

Another true story: on meeting a new team, I was amazed a team's 30+ item working agreement. I remarked that it must have taken a long time to co-create such a comprehensive document. "Oh no," someone replied, "we just went around and collected other team's agreements and put them together!". As you might expect, this team's working agreement held absolutely no meaning for them since they didn't actually generate the content themselves.

## Why does my team need a working agreement?

High-performing teams don't magically form themselves. Figuring out how intelligent, passionate people will do great work together takes deliberate effort. It's worth investing a little time in collaboratively creating team working agreements in order to help the individuals on the team to align around shared expectations. Identifying common values and expected behaviours lays the foundation for building the sense of psychological safety that enables great teamwork.

When I meet with a new team, I often find that their agreement (if there is one at all) focuses on logistics like meeting times and information storage rather than how to *be* together. And there's usually no provision for what to do when someone breaks the agreement, which lessens its power – an unenforceable agreement is no agreement at all.

A strong team working agreement:

- Is collaboratively created by the team itself so that they are holding themselves accountable for doing the things they've identified as important.
- Clearly states what behaviours team members expect of each other: by making expectations transparent, individual team members are also encouraged to become more conscious of their individual actions and what impact they have on others.
- Describes how the team will handle instances where the agreement is brokenso that team members have more confidence about speaking up when this happens: "Hey, we agreed to do X, and that doesn't seem to be happening right now? How do we want to address this?"
- Focuses on a few specific behaviours the team wants to pay attention to. 4-6 items is usually plenty.
- Gets updated as the team evolves – this is a living document

## How can my team build a powerful working agreement?

There's a number of approaches for having the conversations required to build an effective working agreement – choose facilitation techniques that will encourage everyone in the room to talk. I'm a big fan of using LEGO® SERIOUS PLAY® because it's a compelling facilitation technique for cutting through the conversational cruft and getting to the heart of a conversation. (Mike Bowler and I offer a workshop on using LSP to build working agreements – the handout with instructions is downloadable from Mike's website).

Regardless of the facilitation technique used, there's a set of questions I like to use as the foundation for a conversation:

- **Describe the best or worst team member you’ve worked with**.
-
    - Give participants the opportunity to describe the kinds of behaviours they’ve found helpful or destructive in the past – this is useful information for shaping the kinds of interactions we want to foster in our working agreement
- **What’s the superpower you bring to this team that others might not be aware of?**
-
    - Sometimes it’s hard to talk about what we’re good at, especially when it’s a ‘softer’ skill.  Astute teams might also notice if they have a diversity of superpowers or if there’s an imbalance of strengths in certain areas
- **What kind of help do I want from my team to be high-performing?**
-
    - Given how hard it can be to talk about what we’re good at, it’s really difficult for many people to ask for help, especially when the team is new and we’re uncomfortable with vulnerability.  By practicing this at the outset, it lays a foundation for asking for help down the road.
- **What colour signifies conflict to you? Why?**
-
    - This is a really simple exercise that yields very rich conversation about a sensitive topic.  You can use paint chips, crayons, squares of construction paper or any tokens that give people a range of colours to choose from.  Give them the tokens, ask them to pick a colour, then invite everyone to share their answer to the question (And no, not everyone will choose red)
- **What should we do if it seems like one of us is letting the team down?**
-
    - Possibly the most important question as it helps the team prepare for when someone breaks the working agreement.

With the answers to these questions in mind, I ask each team member to propose a behaviour to include in the working agreement.  I remind them to focus on capabilities the team needs to work on, not behaviours that are already solidly established.  And, I encourage them to identify positive behaviours rather calling out undesirable actions (e.g. “do bring team issues up as soon as you sense there’s a problem”, not “don’t have backchannel conversations about problems”).  I also remind them to include an item about how to handle it when the agreement is not being honoured.

Real-life examples of working agreement items I’ve seen teams include:

- When you think someone is struggling, offer help right away – don’t wait
- In team discussions create space for everyone on the team to speak
- Show respect for others by being on time for team meetings (this is a perennial favourite!)
- If you have a problem with someone, speak to them directly before bringing it up with someone else

If the team is small, they might include all of the proposed behaviours in the initial agreement; larger teams will likely need to prioritize which items to focus on for now in order to keep the list focused.

### Care and Feeding of Your Working Agreement

Keep your working agreement visible to the team – if it’s locked away in an electronic oubliette, it’s much harder to refer to when needed.  I encourage teams to post their working agreements as part of their visual management system, so that it’s on hand during team events.

Team working agreements will change as the team evolves and its circumstances change. Review the working agreement periodically to make sure it covers the actions the team needs to work on – this can be a great retrospective discussion.

***

To read this article online with embedded hypertext links, go here:

**https://masteringtheobvious.wordpress.com/2018/04/04/building-alignment-team-working-agreements/**

# About Ellen Grove

**Ellen** Grove is a business agility coach, trainer, and organizational change agent with Agile Partnership. Based in Ottawa, Canada, Ellen works with organizations of all sizes in Canada and abroad to help them put Agile thinking and being into action throughout the organization. Whether working with C-suite leaders or managers or delivery teams, Ellen favours experiential approaches for getting people to talk with each other about the things that really matter so that they can do great work together. She organizes Agile events locally and internationally and she presents at conferences around the world. In addition to a chain of agile-related certifications, Ellen is a an Open Space facilitator, a StrategicPlay® certified facilitator in using LEGO® SERIOUS PLAY® methods, an ORSC-trained organizational coach, and a Training from the Back of the Room trainer.

***

# Leaders of chaos. How traditional management perpetuates chaos in digital IT

By Philippe Guenet

Posted on September 7, 2018

Source: Pixabay

Despite the development of agile ways of working, continuous delivery and product orientation, many companies are still working using the format of projects. In all reality, projects will never completely go away. In the larger enterprise, there will always be a need to coordinate a more significant effort across teams. Projects live and breathe to the beat of the RAG (Red Amber Green) status, like the traffic at a busy junction moves to the rhythm of the lights. Green, all is good. Amber, we need to watch out, maybe ask the team to provide an extra weekly status report (as if extra bureaucracy was going to help them). Red, hell breaks loose, internal audit commands actions and management steps-in to coordinate the crisis.

> Most corporate organisations care about IT delivery when the RAG flags Red. Digital leadership stems from taking an interest in continuous improvements

As a delivery manager by trade, supplying services to corporate organisations, I have seen this pattern repeat over and over again. The management is too busy to care for a Green status. The team identifies possible improvements, and they get rebuked for not focusing on delivering more functionality instead. The team identifies the weak signals of the emergence of systemic challenges, and they remain ignored until a milestone is missed. Once the status is of the appropriate shade of Red to attract the needed attention, it is all hands on deck. Firedrill, all the parties are rallied to the table, at last, the systemic challenges get some attention, and the crisis eventually gets resolved. I have to hold my hand up, I used the Red status a few times when I knew that the systemic issues would not get any attention otherwise. I was discussing this matter with a senior programme manager recently, in an organisation starting with their agile effort. The programme manager was telling me how effective their collaboration was in crisis mode, and he had been through a few incidents lately. I was pointing out to him that good agile leadership was about making such collaboration happen in everyday work, without breaking a sweat.

## Chaotic and the Cynefin framework

The Cynefin framework from Dave Snowden is a good start to explain the cognitive drivers linked with the RAG status. Cynefin is a SenseMaking framework, which defines five domains. The starting domain is Disorder. In Disorder, we have not yet established how to categorise the nature of the challenge, and consequently, have not established the right course of action. There are then two ordered domains, which are characterised by an established causality between action and effect. This causality may be of an Obvious nature or a Complicated nature. Cynefin defines how to manage under those conditions:

- In the Obvious domain, we sense the condition, we categorise it and apply the adequate response, generally policy based.

- In the Complicated domain, we sense the condition, analyse it, calling in the experts if necessary, and respond with the right pattern for the problem.

Mirroring the ordered domains, Cynefin establishes two unordered domains of Complex and Chaotic. In the unordered domains, the causes and effects do not appear connected, at least apriori. It means that:

- In the Complex domain, we probe through experimentations to determine the right path. We sense the reaction to the probes and amplify what appears to work best.

- In Chaotic, we take action to come out of the undesirable situation as soon as possible by setting a direction (good or bad). There is urgency. We then rethink the course of action once the crisis is contained.

## Chaotic, the land of the heroes

The Chaotic domain is very much the context where managers lead from the front, and it reinforces a command style. Sure enough, this is a good feeling when you are a manager. Like captains on the battlefield, you have to keep your wits, think quick and direct action. It is very satisfying driving a team and a project back from the chaos and into normal again. Leadership feels in charge through chaos. Heroes get made from solving chaos. Managers bring all the people around the table, pull the strings that only rank would allow to, trump other demands by pulling more strings with the hierarchy. They ask the team to put in a Herculean effort, hail in praise those that dedicated all-nighters and weekends to the cause. They make sure that they get themselves noticed by the senior management through the whole ordeal too. One would think that going from one crisis to the next would not ring well, but it does. It is a sign of not shying away from a challenge and battling your way through it, with an unquestionable commitment and sense of duty to the business. Organisations do like heroes. The ones that grit their teeth long enough through such situations earn their medals and get the promotion. Yes, let's spell this again, the management that leaves situations to rot to the point of creating a crisis and then proceed to burn-out their people to come out the other side, are the ones that eventually get promoted. The ones that kept their project Green, on the other hand, tend to be regarded as not sufficiently battled-hardened for the next role up. As leaders tend to identify their seconds using patterns they rode themselves, the situation perpetuates and gets engrained into the culture.

> Though the world loves heroes, the agile organisation does not need them!

With this in mind, is it so surprising that organisations have cultivated a dysfunctional relationship with their IT department? As technology is ever more present in the product/service mix of most organisations, businesses on their digital journey have to take this situation very seriously, or they'll only scale the chaos (and many do indeed!). Progressing through cycles of chaos is not sustainable. Businesses operating in this fashion are left exposed to considerable risks and tend to go from one crisis to the next. In the process, they burn-out the talents that they need to retain.

## How do we turn such a culture around?

We are quick to critic the middle management for their behaviour and contribution to the problem. Blame is rarely constructive toward an answer. Antagonising managers results in camped positions and status-quo. The middle management also holds the keys to transforming the ways of working and, for most, are capable of changing. Unfortunately, most organisations give them no reason to. What is the reward for running a more efficient IT with a Green RAG? Less budget for next year, fewer people and a "try harder" from the boss.

Turning this state of play around is actually where successful Digital change starts:

- Systemic management

- Culture of continuous improvements

- Organising for excellence

## Systemic management of the teams

Agile Thinking insists on the need for good collaboration, rightly so. The digital corporate is also aiming to replicate the organisation model of the internet unicorns with flatter structures and autonomous teams, in favour of faster decision making. However, most still manage teams as a collection of individuals and focus on tasks. Can't we see the paradox in this?

If we manage from the individual dimension, we need micro-managers to organise and coordinate the tasks. Leaders justify their worth, insisting on being at the cross-road of every decision and becoming a bottleneck to progress. The team members surrender their autonomy and collaborative incline to the leader on the basis of once bitten, twice shy. Without an approach to managing systemically, teams cannot move to the desired levels of autonomy and collaboration.

Systemic management starts with the managers connecting with the team where the work happens (i.e. in the team's space, not in the manager's glass office). This concept has existed for many decades with the Gemba. Gemba is a term from the Toyota Production System. Gemba consists in visiting the place where the work happens. It is evident in manufacturing, thanks to the physicality of the production chain, but less so in a digital team, where all is very virtual. Going to the Gemba in IT reinforces the need for radiating information in support of visual management. Flow and quality get progressively plastered on the walls (or online boards for distributed teams), and teams can discuss improvements. Managers also need to be equipped with skills and techniques to work with the team as a whole. Hearing the Voices of the System (VoS) generates transparency and highlights systemic challenges early. In turn, problems are anticipated in favour of a better flow of work. By using such practices, the teams become more autonomous, and leaders and teams are at one, with fewer needs for intermediary micro-managers.

## Continuous Improvements (Kaizen)

In the Cynefin framework, Dave Snowden draws a little sign on the limit between the Obvious and the Chaotic domains. This sign indicates a cliff of complacency. When an organisation relies too much on established procedures, it becomes exposed to disruption. When disruption happens, it falls into chaos. As mentioned above, the same happens to our projects. I even got told this week that this had a name: "Watermelons Projects" (All credit to Jay. Thanks!). Green on the outside, Red on the inside. It is not an IT-specific problem either. We heard this week that, a few months away from launch, Crossrail got delayed by a year!  Not a few weeks, but an entire year. We are not monitoring the signals on the inside, so the bad news bubble-up and end-up bursting in spectacular style.

To avoid complacency (and drive continuous performance), the Toyota Production System used Kaizen, a process for continuous improvements. Part of everybody's job is not just to do your job, but to figure out ways of improving how you do it and improving the product-outcome of it. This constant effort is an excellent way of stimulating brains, especially as work on a production chain can be dull. Chihiro Nakao, the father of Shingijutsu Kaizen, was insistent on driving simple improvements that did not need investments. One of his objectives was to stimulate people. Stimulation prevents complacency. Good Kaizen also creates a different dynamic between the leadership and the teams. The teams develop autonomy, self-reliance and collaboration, while management becomes more about facilitation/coaching and inspiring creative thinking. Would it be that Toyota invented 40 years ago what we are trying to achieve today in IT?

Many are quick to point out that manufacturing production is not the same as IT. It is true. Continuous improvements are much easier to experiment within the virtual environment of IT. IT workers are also much more natural knowledge workers than manufacturers on a production chain. There is no excuse really. Only that corporate IT has been too busy tracking Red RAG statuses, that we forgot to implement the discipline of continuous improvements on Green RAGs. Time for a reboot.

## Organising for excellence

No organisation objects to driving excellence, yet very few genuinely do so in IT. The DevOps movement has contributed to bringing the topic onto the agenda, but the effort remains contained into IT. Quality is everybody's job, starting from the demand initiation in the business. Lack of quality is easy to spot on physical goods, but harder on virtual software. You can get some idea from the usability of the front-end or testing some edge cases, but those are only the tip of the iceberg. What quality of code lurks below the surface? When will this come back to bite you? The RAG status does nothing to fix this, in fact, it generally contributes to worsening it. I had a project some time back, which was fluctuating between Amber and Red. The teams were under constant pressure from business and management to hit the deadline. The only part that could give was quality. Did anybody care? The team did, but nobody would listen. So, we got the team to start capturing wasteful practices and logging technical debt, for anything that would not meet the standard that they would expect of their work. The team felt better for it, information got radiated, and it became too inconvenient to ignore. It should not have to come to this point however. IT leadership and product managers should have a vested interest in producing quality. Working from projects does not promote such behaviours. Projects are by nature transient, and the ones that are delivering are rarely the ones that are maintaining. Quality, sustainability, maintainability, rarely registers very high in day-to-day decisions. The short-term attitude brought by a project focus does not support the development of a solid IT foundation for Digital. Effective Kaizen is also unlikely to happen unless there is a long-range commitment from the team and leadership. Beyond the better alignment with customers, the product/service based alignment enables to maintain long-lived teams and leadership. It enables investing in people, in their understanding of engineering and in raising quality in the prioritisation decisions. Such alignment is however mostly orthogonal from how traditional organisations have been set-up (inherited by economies of scale over economies of flow) and constitutes a major systemic challenge in making progress towards Digital performance. We will cover this topic another time. The digital organisation will be the topic of our November Digital Leadership meetup as well.

In summary, the dysfunction of IT in many corporations has been mostly self-inflicted as a result of project-based work and driving the wrong incentives in the leadership. It has promoted behaviours that are incompatible with building the foundations for successful Digital businesses. Solving the situation requires small steps for leaders and a giant leap for leadership (I borrowed this sentence… :-) ). Digital change starts as simple as working with the team to understand how we can make tomorrow better than yesterday: Smarter processes, more autonomy, better products/services, better technology. It is, however, a fundamental departure from an IT leadership based on accountancy over project budgets to an IT leadership enabling product and engineering performance: Systemic management, continuous improvements of products/services and processes, excellence of engineering, and resolving the systemic impediments that stand in the way.

***

To read this article online with embedded hypertext links, go here:

**https://www.henko.co.uk/insights/leaders-of-chaos-how-traditional-management-perpetuates-chaos-in-digital-it**

# About Philippe Guenet

**Philippe** has been involved with Digital change from the early days of the Internet (nearly 25 years) and has led the initiation and delivery of significant change initiatives for large corporate clients & Banks.

Philippe has founded Henko and is now working as an independent Coach/Consultant, or “Coachulting™”. Philippe is using innovative methods to help businesses shape, plan and deploy their digital strategy, organisation change, lean-agile operating model, and most importantly develop their people and leadership. Philippe is developing Digital Thinking and Digital Leadership programmes to skill graduates to Digital and reskill existing workforce and Leaders. Philippe works equally with Corporate clients and Start-up businesses.

Philippe is a certified systemic coach (ORSC), a complexity & systems thinker, a seasoned delivery manager, and a Lean-Agile practitioner. Philippe is the founder and organiser of London’s Digital Leadership meet-up (>2,300 members).

pguenet@henko.co.uk
www.henko.co.uk
www.lead-digital.org
Twitter: @HenkoPhil

# Lean Economics 101: The Power of WIP Limits

By Mike Hall

Posted on March 13, 2018

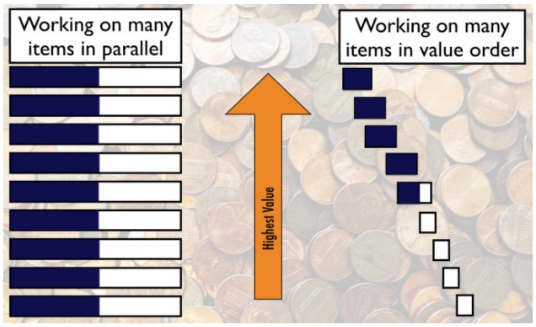
*This is the first in a series of blogs on the topic of Lean Economics, emphasizing the economic aspects of product development.*

I have a friend who once emphatically stated "Agile will never work at my company because we have more projects than people! Everyone here works on 5 or 6 projects at the same time. It's important to show some progress on all efforts to keep everyone happy." Do you have firsthand experience being in this situation? I'll share my response to him at the end.

The concept of Work In Process (WIP) – also called Work In Progress – comes from the world of Lean manufacturing. In Lean manufacturing, WIP is considered a form of inventory – the materials that are currently being worked. As such, it is actually considered "waste" because it ties up cash. That's right – WIP is waste! It needs to be reduced.

## Effective Use Of WIP

Effective use of WIP is when the team takes on a small amount of work, focuses on it, and works it to completion in a short period of time before pulling any additional work. This approach enables a consistent flow of value from the team (as indicated in the diagram below) and is considered much more desirable than context switching across multiple efforts in parallel.


Working on many items in parallel
Highest Value
Working on many items in value order

This image is the property of Agile Velocity.

## Warning Signs You May Be Ignoring Lean Economics

Warning signs that you might not be considering the *economics* of product development include:

- Many things started, but very little finished recently
- Heavy multitasking
- Costly context switching
- Slow, painful deliveries
- Lack of accuracy when predicting delivery scope and dates
- Low product quality and constant rework

All of these add up to a very frustrated team overloaded with work that just seems to never get completed. Reliance on “heroes” is the norm. Overtime is rampant. Customer satisfaction is dropping. Stakeholders are screaming. We scream back: “We need more people!”, or better yet “Please just let me focus on one thing at a time!”

## What is a WIP Limit?

> *The amount of work occurring at any one time is “limited” to what the team can accomplish in a short period of time and established by a constraint called the WIP limit.*

A great description of the economics of WIP is described in Donald Reinertsen’s seminal book The Principles of Product Development Flow. At the time of this writing in 2008, less than 1% of all product development teams were managing WIP constraints. With the success of Agile methods and a renewed focus on the economics of product development, this number has improved greatly over the past years.

Most teams use a default WIP limit according to the available capacity of the team and what makes sense for the effort at hand. Development teams can set their WIP limits for the number of user stories (or tasks) in work at any one time. This WIP limit can provide opportunities for pairing and cross-training.

A Program Management team may have a WIP limit for the number of features being researched, the number of features ready for development, and the number of features currently being deployed. As you move into higher levels of planning in the organization, from individual programs to whole portfolios of programs, executives and business owners may have a WIP limit for the number of business cases being developed, the number of initiatives being considered for funding, and number of initiatives in development at one time. This level-by-level WIP approach helps to create an *alignment of available capacity to the demand*, and also provides a very important visible indicator as to what initiatives are truly being worked on and even more importantly, which ones are not!

Who sets the WIP limit? WIP limits are determined by the teams involved in the work at their respective level (team, program, or portfolio). Thus, team members at each level need a basic understanding of Lean economics and how adjusting the WIP limit can impact the flow of value – both positively and negatively.

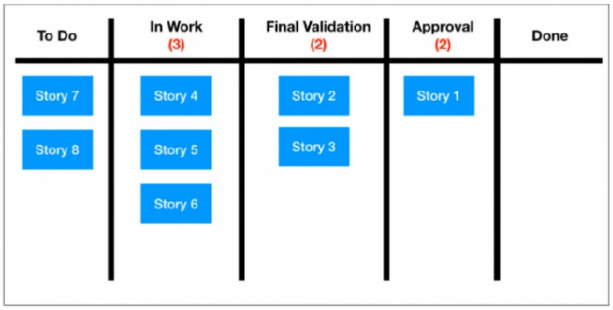
## What is the Right WIP Limit?

A WIP limit that is too small will result in idle workers and minimal flow of value. A WIP limit that is too large will result in delayed delivery and increased average cycle time (time from work starting until done). A WIP limit that is “just right” will result in optimized value flow and predictable delivery cycles. Often times finding the optimal WIP limit takes a bit of trial and error.

WIP limits can be adjusted from time to time in an experimental attempt to drive even higher levels of value. For example, an Agile team of 7 developers may adjust their “in work” user story WIP limit from a default of 7 down to 3 to ensure that a healthy amount of “swarming” is occurring on the highest priority user stories committed for the iteration. This supports the concept of demoing stories to the Product Owner as early in the iteration as possible and the possibility of continuous deployment. WIP can also be adjusted downward to account for the typical vacations and personnel outages that teams experience.

## Visualizing WIP

WIP limits can be applied as work flows through the different stages of development. A common team-level story-board example follows with WIP limits indicated in the appropriate columns. These WIP limits cannot be exceeded, forcing team members to consider helping others in pursuit of the larger goal of getting a user story to Done over beginning a new one.



Teams at the program and portfolio levels use the same technique with their specific column names and WIP limits.

## What’s the Big Deal?

Applying WIP allows us to “stop starting, and start finishing”. Some advantages of WIP and the use of WIP limits include:

- Improved flow of value through your team pipeline
- Productivity gains due to less context switching
- Ability to focus
- Identification of bottlenecks and waste – leading to subsequent remedy
- Improved cycle time
- Promotion of cross-training and less reliance on “heroes”
- Encouraging a focus on quality!

## Conclusion

Now, regarding my friend’s comment about working 5 or 6 projects at once – I used the left hand portion of the first figure (“Working on many items in parallel”) to explain that his approach was making some progress on many things, and every time period check would hopefully show an adjustment in the horizontal progress bar. I deemed this approach “moving yardsticks” instead of truly completing work. In the end, all (yes, all) of his project stakeholders will end up disappointed to varying degrees. I then explained the tremendous cost of context switching across multiple efforts (ref: Quality Software Management: Systems Thinking). I also pointed out that his leaders were suffering from an unwillingness to make tough prioritization decisions, and that this often requires a cultural change at the leadership level – stay tuned for future blogs in this series discussing these concerns.

After a few more (lite) beers to let it all sink in, I explained the good news – there actually is a much better way! Organize all of the work into a single team backlog. Prioritize the work such that single features/stories can be worked to completion and context switching is reduced. Apply WIP limits at all levels (team, program, and portfolio) to ensure a healthy flow of value through the team’s work pipeline. And then say goodbye forever to unmet expectations!

***

To read this article online with embedded hypertext links, go here:

**https://agilevelocity.com/lean/lean-economics-101-the-power-of-wip-limits/**

# Lean Economics 101: Parallel Development Is Killing Your Productivity!

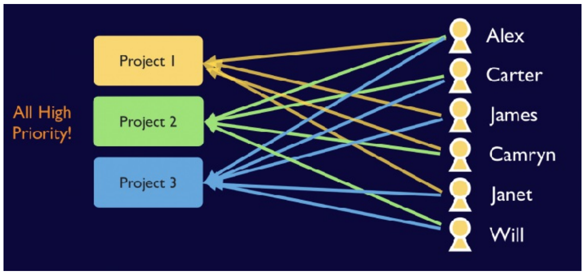
By Mike Hall

Posted on April 9, 2018

This is another in a series of blogs on the topic of Lean Economics, emphasizing the economic aspects of product development. You can find the first post about WIP limits here.

As a developer, do you work multiple projects in parallel? As a leader, do you have a team which works multiple features in parallel? Is there more work than people?

Parallel Development is working multiple projects or features at the same time, as shown in the diagram below. It is a common situation I often see when coaching or delivering training. Parallel Development has, unfortunately, become commonplace in industry and accepted as the default standard way of organizing work as our work lives become busier and busier–just assign more and more projects to people, and hope they figure out how a way to get it all done!



Often times when I query for reasons behind this type of work model, the refrain is a familiar “We have more work than people. Thus, our people need to be good enough to work multiple projects and show good progress on all of them.”

## Parallel Development: The Efficiency Killer

There are many disadvantages to Parallel Development including:

- Delayed delivery of business value
- High cost of context switching
- Inefficient work model
- Low employee morale

All of this can lead to a frustrating work environment where employees are routinely interrupted to begin work on a new project with no regard to their current workload. Everyone is 150% (or choose your number) allocated piecemeal across multiple projects. If the feeling of "starting is common, but finishing is uncommon" is pervasive, then your current work model is economically challenged and likely needs a major adjustment.

In a Parallel Development approach, the business value of each feature is delayed as the team members routinely multitask by working on multiple projects and features as shown in the figure above. *Time lost in context-switching can be more than time applied to business value*. Often in this situation, the desire to "show some progress" to all stakeholders is deemed more important than delivering value at the earliest point in time.

## Serial Development: A Better Way!

The good news is this: there is a better way! Serial Development is when the Agile team focuses on a single project and single feature at a time.

Advantages of Serial Development include:

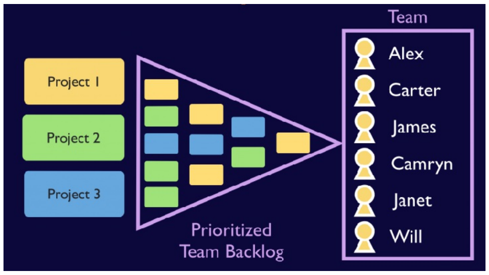
- Consistent, early delivery of value
- Limited context switching
- Focused development work
- High levels of employee satisfaction

**Serial Development requires courage!**

It requires the courage…

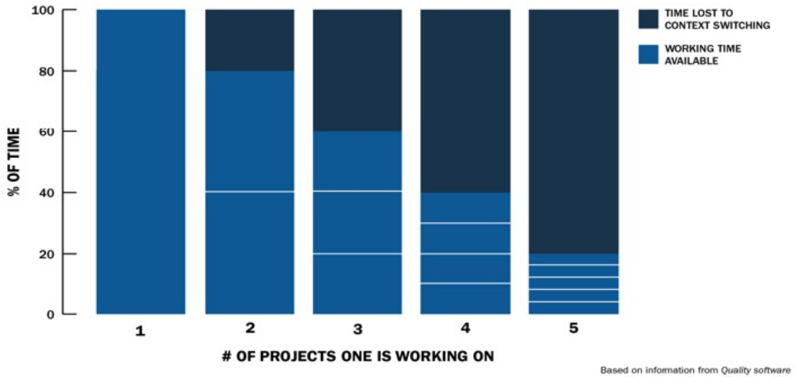
- to ruthlessly prioritize features–not only within the same project but comparatively across multiple projects as shown in the figure below.
- to allow the team to focus their development efforts on one feature at a time.
- to inform your stakeholders as to when their features will likely be developed instead of "We have started, but not much progress yet".

And Serial Development requires courage in the form of coaching your stakeholders that Serial Development will help ensure that their feature(s) gets the development team's attention at the absolute earliest opportunity based on prioritization.



## Comparison: Context Switching

Not convinced? Let's compare Parallel Development versus Serial Development in terms of context switching. Parallel Development experiences significant context switching costs as the team members bounce back and forth across multiple features and projects trying their best to keep everyone happy. The generally accepted authority on the cost of context switching comes from the seminal book Quality Software Management: Systems Thinking by Gerald M. Weinberg.



Based on information from *Quality software Management: Systems Thinking*, p284

In summary, **if you are working on 5 projects in parallel, you are only 20% productive!** Moving over to a Serial Development model will help you gain back the 80% productivity lost in context switching. Multiplying this by all members of your team can result in significant levels of productivity gain for the entire team!

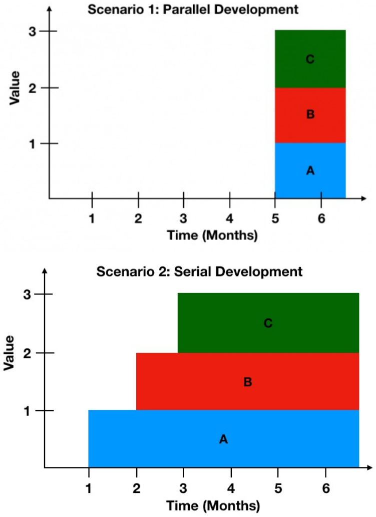
## Comparison: Value Delivery

Now let's compare Parallel Development versus Serial Development in terms of delivery of value. I'll use a simple example from one of our training courses. This example is feature-based but also applies to working multiple projects.

*Setup*: You work on a team that has a team backlog consisting of 3 features: A, B, and C. Each feature takes the entire team 1 month of time and delivers 1 unit of value. Let's investigate the value graph across two scenarios – Parallel and Serial Development.

*Scenario 1*: Parallel Development with an assumed 40% context switching overhead (ref: Quality Software Management: Systems Thinking). The team multitasks across all 3 features making some progress each month (approximately 20% complete) while experiencing the context switching cost. At the end of month 1, no feature is completed. At the end of month 2, no feature is completed. Same for month 3, and same for month 4. Finally, at the end of month 5, the team delivers all 3 features.

*Scenario 2*: Serial Development focusing on 1 feature at a time. At the end of month 1, feature A is delivered. Context switching cost is minimized and occurs only during the planning associated with kicking off work on the new feature, e.g. during sprint planning and backlog refinement. The team delivers predictably, early and often, 1 feature per month. Delivered features continue their value month-to-month as the next feature is being worked on.



Comparing the two value graphs, which is better in terms of value delivery and cost of delay? In the Serial Development approach, the customers not only enjoy the incremental value of the features delivered as soon as possible, but they receive all 3 features in total much earlier than in Parallel Development!

## Comparison: Revenue

As a final comparison, let's look at a simple revenue example. In the above graphs, assume each feature delivers $100,000 of revenue each month.

For Parallel Development, the total cumulative revenue at the end of each month is:

- Month 1: $0
- Month 2: $0
- Month 3: $0
- Month 4: $0
- Month 5: $0
- Month 6: $300,000

For Serial Development, the total cumulative revenue at the end of each month is:

- Month 1: $0
- Month 2: $100,000
- Month 3: $300,000
- Month 4: $600,000
- Month 5: $900,000
- Month 6: $1,200,000

For Serial Development, the total revenue at the end of month 6 is 400% greater than Parallel Development. *The bottom line for this example – developing features serially delivers the business $900,000 in additional revenues!*

## Four Steps to Begin Serial Development

1. Ruthlessly prioritize the feature requests across all project demands.
2. Group the user stories in your team backlog based on a single feature for a single project.
3. Pare the stories down to a minimum marketable feature that can be launched. Ideally, the team will be able to complete the highest priority minimum marketable feature in the next sprint (or two).
4. Encourage swarming within the iteration to focus on a few high-priority user stories and complete them as early in the sprint as possible. Effective use of WIP limits can lead to swarming

Stop the madness! Parallel Development is wrought with waste and creates a chaotic environment for employees. Serial Development is a much more economical work model ensuring the earliest delivery of value to the end customer and higher employee morale. If your team members are currently mired in Parallel Development, then take action to shift over to a focus-based Serial Development approach. This may take a bit of courage and hard work, but the benefits are so well worth it.

***

To read this article online with embedded hypertext links, go here:

**https://agilevelocity.com/lean/lean-economics-101-parallel-development-is-killing-your-productivity/**

# 5 Practices To Start Scaling Agile

By Mike Hall

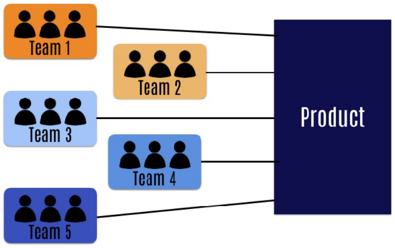
Posted on June 4, 2018

When considering scaling Agile, it is important to start small. Some of the various scaled Agile frameworks available can look incredibly complex, while others can look simplistic and incomplete at first glance. Relax. Don't overcomplicate it. Start small with these proven best practices—and then fill in the rest as your Agile journey continues!

## Situation

How do you know when to apply Agile at scale? Before we talk about scaling, let's determine if you truly need to scale. Common situations causing a need for scaling include:

- More and more large initiatives that require multiple teams
- Limited collaboration between teams working on similar or dependent efforts
- Late-breaking dependencies that impact the delivery schedule
- Teams focused primarily on their deliverables, often at the expense of the larger program objective
- The pursuit of full enterprise agility beyond just team-level agility
- Burning platform—current way of doing business is inadequate to survive



Many enterprises have experienced success with Agile at the team level. Their teams are working iteratively and incrementally, delivering some form of business value every 2 weeks. But then they realize that Agile is also needed for their initiatives that are broader in scope and require a significant number of teams.

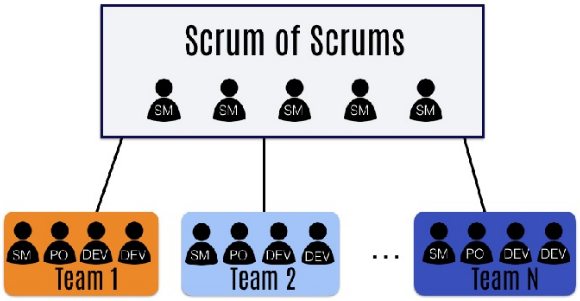
In this situation, how do we ensure that the multiple teams involved are fully synchronized and working effectively towards a common goal? How do we ensure that they are managing risks well? How do we help ensure the success of the program?

We recommend the following five practices to start scaling Agile:

- Scrum of Scrums
- Product Owner Sync
- Program Backlog
- Dependency Identification and Scheduling
- Common Integration Environment

## Scrum of Scrums

The Scrum of Scrums is a scaled version of the team's daily standup. The purpose of the Scrum of Scrums is to ensure alignment of all teams working on the program. Most Agile scaling frameworks (SAFe®, Nexus, Scrum at Scale, etc.) recommend this synchronization event.



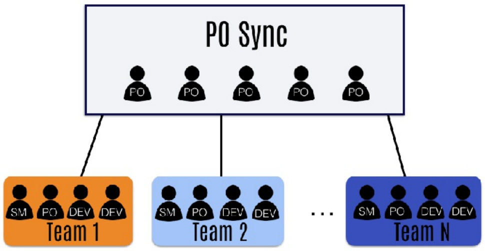
The Scrum of Scrums is held 1 – 2 times per week for 15 – 30 minutes. Each team sends a representative, typically the ScrumMaster, to the Scrum of Scrums. If additional expertise is needed for a discussion, the ScrumMaster can bring in development team members to assist in understanding and decision making. If the multi-team program has a Program Manager, the Program Manager also attends to better understand the inter-team risks.

Topics discussed at the Scrum of Scrums include:

- Impediments, especially program or organizational level
- Inter-team dependencies
- Upcoming milestones
- Scope and schedule adjustments
- Integration progress
- Looming risks
- Action items

## Product Owner Sync

The PO Sync is the content-focused equivalent of the Scrum of Scrums. It is a practice performed even in smaller organizations working on the same platform. The purpose of the PO Sync is to ensure alignment of the product vision and work-related content across all involved teams. SAFe® and Scrum at Scale scaling frameworks recommend this synchronization event.



The PO Sync is held 1 – 2 times per week for approximately 30 minutes. Each team sends their Product Owner to the PO Sync. If the multi-team program has a Program Manager, the Program Manager also attends to ensure the team efforts are cohesive and supportive of the program goals.

Topics discussed at the PO Sync include:

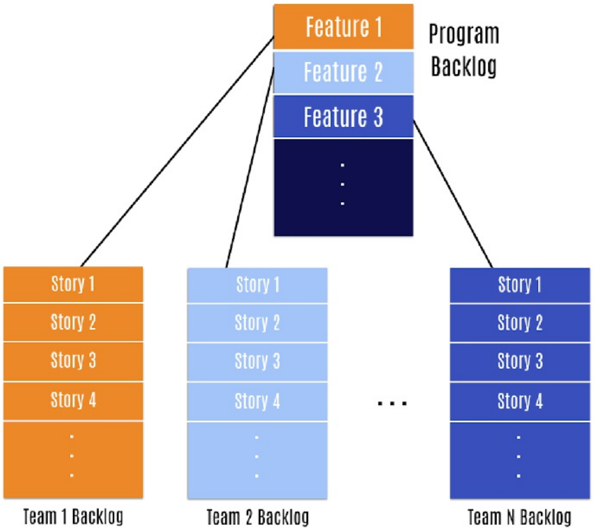
- Functionality, performance, and interoperability
- Scope and schedule adjustments
- Feature prioritization
- Validation of feature prioritization within the Team Backlogs
- Coordination and scheduling of inter-team dependencies
- Team progress on features

- Decomposition of features to fit within a release
- Minimum viable product set and minimum marketable feature set
- Action items

## Program Backlog

When multiple teams are working on the same initiative, it is important to establish the Program Backlog. For our purposes, the term “program” is a coordinated multi-team effort. The Program Backlog is a higher-level backlog different than each team’s Product Backlog. The Program Backlog is created and maintained by the Program Manager role or a Product Owner from one of the teams. The Program Backlog consists of features in priority order. All Agile scaling frameworks recommend this artifact.

Teams take the features from the Program Backlog and decompose them into user stories. The user stories then become part of the Team Backlogs. The order of the user stories within the Team Backlogs should reflect the feature priorities. Care should also be taken to allow each team to focus on one feature at a time as serial development is much more economical than parallel development.

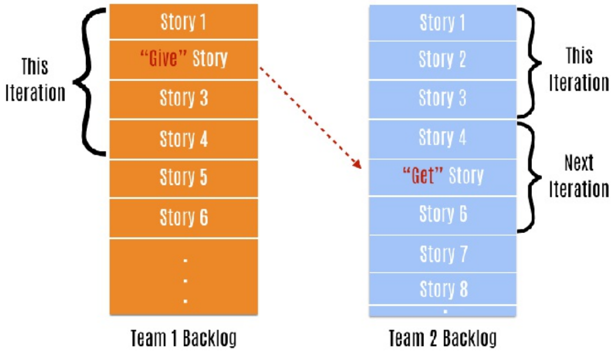


The teams then develop the user stories comprising the feature across iterations. In the simplest approach, a feature is developed by one team. In a more complicated approach, a feature is developed by multiple teams, with each team developing their specific user stories while collaborating tightly with the other teams working on the same feature.

## Dependency Identification and Scheduling

One of the biggest risks when scaling Agile is inter-team dependencies. To mitigate the risk associated with dependencies, teams working on the same program should plan together (iteration planning and/or release planning) to identify the dependencies between teams. Dependencies can also be removed or reduced by adjusting team composition to be more cross-functional.

Dependencies are bi-directional in that there is both a giver and a getter. To ensure visibility, each dependency should be tracked within the Team Backlogs as either a “Give” or a “Get”.



A "Give" is where one team provides a specific deliverable to another team *on or before a specified date or iteration*. A "Get" is where a team *receives a specific deliverable from another team by the specified date or iteration*. A "Get" user story will typically involve integration activities across teams.

Care should be taken by the giving team's Product Owner to prioritize the "Give" in advance of the recipient team's "Get" in order to reduce the risk of developing both the "Give" and "Get" within the same iteration. As a general rule, the "Give" should be provided 1 or 2 iterations prior to the "Get" integration work. Also when integration work is occurring, the "Give" team should allocate some bandwidth to support the "Get" team with any potential issues found during integration.

## Common Integration Environment

Finally, integration is key when multiple teams are developing the same product. You will need to establish a common integration environment for multi-team programs. Ideally, this is done prior to iteration 1 to ensure demonstration of working software from the integration environment each and every iteration.

Typical activities in establishing a common integration environment include server setup, continuous integration (CI) tooling, test case automation, and automated deployment migration.

As in any scaled Agile situation, integration should occur each and every iteration. The true measure of multi-team program progress is the level of integration, not the progress of each team in their local environments. Any approach that defers integration adds an unacceptable level of risk to the program schedule.

Start small and "think Lean" when considering how to scale Agile at your company. Over time, add in the additional constructs, policies, and approaches on an as-needed basis where it makes sense.

Scaling Agile is a crucial consideration of expanding beyond team-based Agile and pursuing enterprise agility. To ensure your success, call Agile Velocity. We have the appropriate experience in the various scaled Agile frameworks (SAFe®, DAD, LeSS, Scrum@Scale, and others) and can help you apply best practices for your specific needs.

If you're specifically looking to start a SAFe® implementation, we recommend starting with Leading SAFe® training. You can explore our public and private training options on our Leading SAFe® training page by clicking **here.**

***

To read this article online with embedded hypertext links, go here:

**https://agilevelocity.com/leadership/5-practices-to-start-scaling-agile/**

# About Mike Hall

**Mike** is a Senior Agile Coach/Trainer at Agile Velocity with over 17 years of Agile transformation experience. He holds many Agile and SAFe certifications, but more importantly has practical experience in leading enterprise scaled Agile transformations (SAFe), Agile process creation for startup companies (Lean Startup), program-level coaching, team Scrum/Kanban coaching, individual mentoring, and developing/delivering Agile training. Mike's background is in software development and leading technology teams. He is also the holder of ten patents issued by the US Patent Office, in areas such as pattern recognition, intelligence engines, and predictive systems.

Mike has been married for 35 years and has three grown children and a dog named Rocco. His hobbies include kayaking, fishing, perennial gardening, and rooting for the Dallas Cowboys.

# Why we don’t re-estimate story points

By David Hawks

Posted on March 26, 2018

To estimate or not? To re-estimate or not? This short article will tackle the second question. We’ll weigh in on #EstimateGate in a later blog post.

Most teams are tempted to re-estimate stories once the Sprint has begun; as teams put fingers to keyboard, it is natural that they learn more about what is actually involved. Hence, the temptation to re-estimate story points because (and this is just a guess based on our coaching experience), they want to be right, or accurate.

## Precise vs. Accurate vs. Consistent

A quick review of terms:

**Accurate** – In terms of measurement, accurate is synonymous with exact. It is not 11:00, it is 11:06am. A common way to describe accuracy is hitting a target on the bullseye.

**Precise** – You need a few more throws at the dart board to be precise as this adjective refers to the closeness of several things in relation to one another. A group of darts in the same area means the thrower is precise. You can be accurate *and* precise if you have a cluster of darts at or near the bullseye. However you were accurate and imprecise if you have darts all over the place and only hit the bullseye once.

**Consistent –** Like precision, you need several measurements to use this term. Consistency describes the measurement of a group of items that are the same. If each apple in a bucket of apples weighs around 5oz, then a farmer can describe that weight to be consistent with apples.

When teams have been working together for a long time, their estimates become consistent. If a team has built one travel app, then they can rely on their past experience to estimate their next travel app. If they have built 10 travel apps as a team, then their estimates for travel apps should be consistent. Using this past information, it is likely that their estimate for the 11th app will be accurate as well as precise or consistent.

Instead of shooting for accuracy, teams should aim to be precise or consistent.

## Fight the temptation to re-estimate

Some teams believe that velocity is a measure of how much they got done. With this mindset, velocity proves their work and effort. Therefore, it is tempting to change the estimate if more work (and bigger size) was needed to finish the story.

However, velocity is a measurement used to determine what teams can get done in the FUTURE.

Teams never have perfect information when they are estimating and planning upcoming work.

If a story was estimated to be an 8 and it ended up being way easier (let's say in the 2 range), don't change it. The past should stay in the past. Anomalies are concerning, only if it happens a lot. We are looking for consistency in story point estimation not accuracy. Also important to note that it's important to estimate the whole backlog with same fidelity (understanding) of knowledge.

Re-estimation of story points…

- Does NOT improve accuracy
- Decreases consistency in their ability to forecast

## So what do you do if the story ends up bigger?

If for some reason the scope changed and the story is bigger, then the Product Owner should create another story capturing the NEW work. This new story should then be estimated and prioritized within the backlog.

If sizing is consistently off, during the retrospective, dig deeper with the following questions:

- Why are we only identifying the size issue at the end of the Sprint?
- What caused the story to be a different size?
- Was it a scope issue or a time issue?
- Did we do something this Sprint that made work easier (and therefore smaller) or harder (and therefore bigger)?

If you have more questions about sizing story points or backlog refinement, register for our next tactical webinar. During this Q&A we'll tackle FAQ's about refining the backlog, including sizing and estimating, plus questions from the audience. We also cover this topic during our Certified Product Owner Workshop or during a session with one of our Agile Coaches.

***

To read this article online with embedded hypertext links, go here:

**https://agilevelocity.com/agile/why-we-dont-re-estimate-story-points/**

# About David Hawks

Founder and CEO of Agile Velocity, **David** Hawks is a Certified Enterprise Coach and Certified Scrum Trainer who is passionate about helping organizations achieve true agility beyond the basic implementation of Agile practices.
David's primary focus is to guide leaders through their Agile transformation by helping to create successful transformation strategies and effectively manage organizational change with a focus on achieving real business results.
He received his Bachelor of Business Administration in Management Information Systems degree from the University of Texas at Austin.

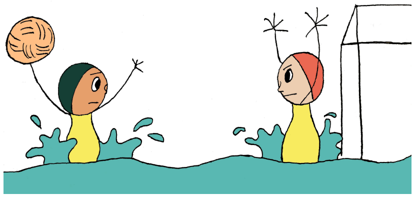
When not helping organizations build lasting success, he can be found swimming, running, biking, spending time with his wife and kids, or supporting his Texas Longhorns. David not only bleeds burnt orange but is known for a crazy, 18-hour tailgate experience, arriving at the lot first and leaving last.

# Becoming a 10x Developer

### 10 Ways to be a Better Teammate

By Kate Heddleston

Posted on March 12, 2018



When I was first learning to play water polo, a coach told me something I've never forgotten. He said, "Great players make everyone around them look like great players." A great player can catch any pass, anticipating imperfect throws and getting into position. When they make a return pass, they throw the ball so that the other person can make the catch easily.

Software engineering today is a team sport; like water polo, you can't build incredible software systems alone. So when I first heard the concept of the 10x engineer, I was confused. How could someone be so talented that it overshadows the power of teamwork? In my experience, individual excellence is necessary, but not sufficient, for greatness. Focusing purely on individual achievement misses the larger picture that teams are required to build great software. So I decided to change the definition of a 10x engineer to this:

## A 10x engineer isn't someone who is 10x better than those around them, but someone who makes those around them 10x better.

Over the years I've combined my personal experience with research about building and growing effective teams and turned that into a list of 10 ways to be a better teammate, regardless of position or experience level. While many things on this list are general pieces of advice for how to be a good teammate, there is an emphasis on how to be a good teammate to people from diverse backgrounds.

## 10 Ways to be a Better Teammate

1. Create an environment of psychological safety
2. Encourage everyone to participate equally
3. Assign credit accurately and generously
4. Amplify unheard voices in meetings
5. Give constructive, actionable feedback and avoid personal criticism
6. Hold yourself and others accountable
7. Cultivate excellence in an area that is valuable to the team
8. Educate yourself about diversity, inclusivity, and equality in the workplace
9. Maintain a growth mindset
10. Advocate for company policies that increase workplace equality

## 1. Create an environment of psychological safety

In 2012, Google launched Project Aristotle to study hundreds of Google teams and figure out why some teams performed better than others [1]. The study found that there were only two key differences between more productive teams and less productive teams. One of those two the key components was something researchers call Psychological Safety. Harvard Business School professor Amy Edmondson described it as a “shared belief held by members of a team that the team is safe for interpersonal risk-taking” and “a sense of confidence that the team will not embarrass, reject or punish someone for speaking up.” [1] Creating an environment of psychological safety creates a space where team members can trust one another and share their opinions and ideas about work freely. Here are some ways to foster an environment of psychological safety within your own workplace.

1. Acknowledge people’s ideas and feelings in a non-judgemental way. Acknowledgment is a separate step from judging or assessing.
2. Respond to ideas with a “yes and” attitude to build off what your teammates say (like in improvisational comedy!) [2].
3. Give people the benefit of the doubt. Believe them until proven otherwise, as opposed to making them prove themselves until you believe them.

## 2. Encourage everyone to participate equally

Project Aristotle, the research study on effective teams at Google [1], found one other important component for productive teams. It’s a phenomenon that the academic world has named ‘‘equality in distribution of conversational turn-taking.’’ Basically, this just means that people on effective teams participate equally. This doesn’t mean that people have to speak equally in every meeting, but that, over time, everyone on a team will contribute equally. So how can you foster a team culture in which everyone participates equally?

1. Ask people their opinions in meetings.
2. Invite discussion with your language by using words like “I think” and “maybe”. (I call this conversational style: "Maybe you should talk more like a woman?")
3. Communicate frequently and informally [3].
4. Notice when someone else might be dominating a conversation and make room for others to speak.

## 3. Assign credit accurately and generously

Giving credit to people accurately for their work is an important part of establishing trust in a team and organization [4]. Many of us have had a personal experience where we felt like we didn’t receive credit for our work, or credit was given to the wrong person.

One of the great things about giving other people credit is that it doesn’t just make the other person look good, it also makes you look good. People who acknowledge others are seen as both smarter and more likable by other people. Basically, it’s a win-win and there’s no reason to hold back in your praise of others.

What can you do to help foster a culture where people assign credit to others for their work?

1. Make sure you take time at the end of each project to thank the people who helped you.
2. Especially try to notice people who are quiet and don’t do much self-promotion, or people who are new and lack confidence.
3. Be honest, specific, and genuine when assigning credit and praising others’ work.

## 4. Amplify unheard voices in meetings by repeating what they say

In 2009, a group of President Obama's female staff banded together and devised a strategy they called "amplification" [5]. They were experiencing a lot of the things women face in male-dominated workplaces—they were having trouble getting into important meetings and, when there, were often overlooked or unheard. So they would amplify each other's voices. When a female staffer made a key point, the other women would repeat it and give credit to the author, which forced everyone to notice where the idea came from. President Obama himself noticed this and made a point to call on more female staffers more. By his second term, there was an even gender split and half the departments were headed by women.

This is such a simple, concrete thing that any person on any team can do for their teammates. Notice when people are overlooked or unheard and amplify what they say. While it's common for women to be spoken over [6], it can also happen to people who are soft-spoken, shy, or introverted.

## 5. Give constructive, actionable feedback and avoid personal criticism

People pretty universally dislike being criticized and, when not given thoughtfully or constructively, criticism can actually damage people's performance [8]. I've talked about this before, but making sure that your feedback is thoughtful and constructive is a really important way you can be a great teammate. Additionally, the way we give feedback can be biased, so spending time learning to give feedback well is important for diverse teams.

For example, there's a common perception that women receive criticism that men don't. Women report being called "bossy", "pushy", and "aggressive" more than their male peers. In 2014, Kieran Snyder, founder and CEO of Textio, decided to test that idea [7]. She collected 248 reviews from 180 people, 105 men and 75 women, and analyzed the contents of those reviews. What she found was surprising, even if you know it's there. Of the women's reviews, 87.9% percent contained criticism or negative feedback as opposed to 58.9% of men. Furthermore, the criticism given to men and women wasn't the same. Of the criticism women received, 76% of it was personal in nature as opposed to 2% of criticism given to men.

Here are some ways you can give better feedback.

1. Ask people if they're open to feedback before giving feedback.
2. Focus your feedback on the person's work as much as possible.
3. Tell people how they can make things better. Clearly identify the issue as you see it and explain how you think the person can make things better.
4. Avoid personal criticism. If you feel that you need to give someone feedback on their person, work with a manager or HR to make sure that you have your thoughts organized.

## 6. Hold yourself and others accountable

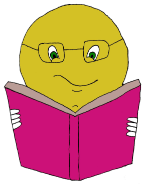
Recently I met with a friend of mine, James, who was a football player in college. He's now the COO of a startup, and he mentioned that he was spending a lot of his time teaching what he considered basic teamwork to employees, especially around accountability. As I thought about it more, I realized that James has spent thousands and thousands of hours practicing being a good teammate. Things that are obvious to him about working well as a team might not be obvious to others, especially when it comes to accountability. Football has a strong culture of accountability, and some of the ways football players hold each other accountable are through being on time to practice, having a positive attitude, encouraging teammates, and holding each other to a standard of excellence. Here are some ways you can hold yourself and others accountable.

1. Get your work done as on-time as you can (I know as engineers estimates are one of our biggest challenges, but smaller projects with more accurate estimates help foster accountability).
2. Jeff Lawson once told a group of founders that the most important thing is “doing what you say you’ll do”.
3. Help others, and ask for help when you need it.
4. For big projects or issues, stay present until your team’s work is finished [9] (whether that presence is in-person or through remote tools like Slack).

## 7. Cultivate excellence in an area that is valuable to the team

When we talk about being a 10x engineer, this is usually what people are referring to—individual excellence. Individual excellence is a necessary part of being a good teammate. After all, you need to do something for your team. What you choose to become excellent at is really about what motivates you; it should be something that gives you energy and fits your skills and interests. Excellence takes a lot of time and energy to cultivate, especially as our individual knowledge base is becoming more and more specialized [10], so pick something you enjoy doing because you'll probably be at this a while. I feel that individual excellence is harped on a lot in our society, so I’ll let you read the slew of self-help books and blog posts out there on how to be more excellent at your craft.

## 8. Educate yourself about diversity, inclusivity, and equality in the workplace



Diversity and Inclusion is a team sport—we need everyone at every level to participate. One of the number one things you can do to be a good teammate is to educate yourself about how gender and race discrimination take form in the workplace. In the same way that you need to stay educated on programming languages and tooling, it’s important to stay up to date on all the amazing writing and research about how to create more egalitarian work environments.

There is a joke that “behind every woke man is an exhausted feminist”, and probably behind every woke white person is an exhausted person of color. Let’s change that. Anyone can read and do research.

1. Read everything
2. Ask people for reading suggestions or join mailing lists
3. Listen as much as you can
4. Your opinion is as valuable as your education, so if you have not educated yourself, your opinion is not valuable.

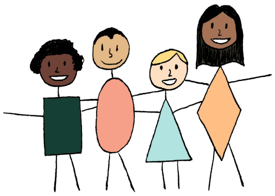
## 9. Maintain a growth mindset

Thirty years ago, psychologist Carol Dweck was interested in the concept of failure and resilience [11]. She and her research team noticed that some students rebounded from failure while others were demoralized even by small setbacks. She wanted to know why. After studying thousands of children, she coined the term Growth Mindset, which refers to the belief that abilities and intelligence can be developed. Students who believe that they can increase their abilities and intelligence rebound from failure, while students who believe intelligence is fixed are more likely to be demoralized by setbacks.

1. Remember you can learn anything given time, effort, and the internet.
2. Be prepared for feedback on how you can improve.

3. There is no finish line; being a great engineer and a great teammate is a lifelong, daily practice.

## 10. Advocate for companies policies that increase workplace equality



Finally, speak up about ways you think your company can create a more egalitarian, inclusive workplace for every member of your team. No matter what your level within the organization, you can advocate for policies that will improve your work environment. Reading and doing research per recommendation #8 will make this step a lot easier. Some examples of organizational changes that have proven effects are:

1. The Rooney Rule: for any key position, you have to interview at least one person of color for the role [12]
2. Evaluate candidates in groups for promotions [13]
3. Making meetings, salaries, key initiatives, and internal processes more transparent.

## Outro

After my own water polo career ended I started coaching, and one of the things I always told the girls I coached was this. Your ability to win is the sum of your talent multiplied by how well you work together as a team.

**Ability to Win = Σ(talent) * teamwork**

Modified for the work world, this equation looks like this:

**Productivity = Σ(talent) * teamwork**

Teams that work well together can outperform groups of people who are collectively more talented because there is a multiplication factor for teamwork. We’ve seen this countless times in sports, technology, and society. It’s not a coincidence that small teams of individuals are able to build incredible software when they work well together. So yes, cultivate excellence in the things you do as an individual, but remember, as my coach told me, true excellence is marked not by how great you are, but by how great you make those around you.

## References

1. Duhigg, Charles. What Google Learned from its Quest to Build a Perfect Team. NyTimes.com, 2016.
2. Edge Studio. Yes and...How Improv Attitude Helps in Many Ways. edgestudios.com, 2015.
3. Pentland, Alex “Sandy”. The Hard Science of Teamwork. Harvard Business Review, hbr.com, 2012.
4. Jain, Sachin H. The Importance of Giving Credit. Harvard Business Review, hbr.com, 2014.
5. Landsbaum, Claire. Obama’s Female Staffers Came Up With a Genius Strategy to Make Sure Their Voices Were Heard. The Cut, thecut.com, 2016.
6. Chira, Susan. The Universal Phenomenon of Men Interrupting Women. NyTimes.com, 2017.
7. Snyder, Kieran. The Abrasiveness Trap: High-achieving men and women are described differently in reviews. Fortune.com, 2014.
8. Nass, Clifford; Yen, Corina (2010-09-02). The Man Who Lied to His Laptop: What We Can Learn About Ourselves from Our Machines. Penguin Group US. Kindle Edition.

9. Gleeson, Brent. Why Accountability is Critical. Forbes.com, 2016.
10. Jones, Benjamin F. The Science Behind the Growing Importance of Collaboration. Kellog Insight, insight. kellog.northwestern.edu, 2017.
11. Dweck, Carol S. Mindset: The New Psychology of Success. Ballantine Books, 2006.
12. The Rooney Rule. Wikipedia. Wikipedia.org, 18 March, 2018.
13. Starvish, Maggie. Better by the Bunch: Evaluating Job Candidates in Groups. Harvard Business School, hbswk. hbs.com, 2012.

***

To read this article online with embedded hypertext links, go here:

**https://www.kateheddleston.com/blog/becoming-a-10x-developer**

# About Kate Heddleston

**Kate** Heddleston is a writer, software engineer, engineering leader, and recovering entrepreneur in San Francisco, CA. She got a master's degree in Computer Science - Human-Computer Interaction and studied Communication - Human-Computer Interaction for her Bachelor's. When she's not building software or writing about engineering leadership, she ocean swims and thinks about all the books she's totally been meaning to read. You can find everything you never wanted to know about her here: [https://kateheddleston.com/about](https://kateheddleston.com/about).

# About GeePaw Hill

**GeePaw** Hill, a writer, trainer, coach, speaker, and above all, hard-core computer geek steeped in the tradition we now call agile software development. His mission is to help organizations that rely on “makers making”.

# Coaching? Like People

By GeePaw Hill

Posted on February 28, 2018

Vimeo video: Coaching Structure Development? Like people!

Hey, there, it's GeePaw! And you are watching the first ever GeePaw video aimed specifically at coaches. I hope you like it.

Do you know what the first absolute prerequisite for being a successful software development coach is? **You have to like people.**

You have to like people. You have to like their style. You have to like their resistance. You have to like when they do the thing you want them to do. You have to like when they don't do the thing that you want them to do. You have to like people.

We get into this business because we're very hot on a system, or a method, or a technique, or a something, and we want to share it with everybody.

But it's really easy to get caught up, caught up in the system, caught up in the method, caught up in whatever flame war is running around the building today. It's important to remember, though, the secret requirement for all successful coaching enterprise, is to like people.

## What I'm Not Saying

Now, before we get very far, we have to address a couple of things I'm not talking about. I am not suggesting that you have to be an extrovert, a person who gains energy rather than losing it by being around other people. You don't have to be an extrovert.

And I'm not suggesting that you have to be some sort of rainbow-farting unicorn floating up in a blue sky with just enough fluffy clouds in it to prove to you that, truly, every day is a beautiful day.

You don't have to be either of those two things. I'm not either of those two things, not even by a stretch. But I like people. I like how they think. I like what they smile about and what they cry about. I just like people. What can I tell you?

And if you're going to be a successful coach, you have to do that, too. Now, listen, I'm a pro. And I always find something nice to say about every team that I work with, because it's a perfectly ordinary and healthy part of the job. I admit, sometimes, it's a really big stretch. **Gosh, you know, you guys just have the best shoes of any team I've ever worked with.**

But truthfully, I like people.

I like knowing how so-and-so met her significant other. I like knowing why so-and-so became a geek. I like knowing what happened at the house last night over dinner that was funny? I like knowing that my friend buys tons and tons of video games but never ever has time to play any of them. All of those things, I like about people.

**I like people because we're so messed up and we try so hard.**

That's my secret to liking people. That's what led me to like them so much.

We're so broken. I'm broken. You're broken. Everybody around you is broken. And yet, we're all trying so hard to be who we want to be, to do the right thing, to be kind, and strong, and fast, and smart.

And that is what let’s me like people so much.

## Play More Favorites

Let me ask you something. Do you ever play favorites? Are you ever a little warmer to somebody else because they’re one of your favorites, a little more tolerant of them, a little more cheerful around them, happy to see them, happy to listen to them, happy to talk about things that don’t have anything at all to do with work?

So do you play favorites?

Good.

Keep it up. Play more favorites. Pick more favorites to become your favorites.

## Remember The Last Time?

Do you remember how it felt when you were listened to, when you were asked, when you got help from somebody else, or you gave help to somebody else, when you were regarded kindly by someone?

Do you remember how powerful you felt, how much stronger, how much more whole, how much more hopeful about what was coming in front of you, and about the possibility of you actually pulling it off?

So. Dig it.

Do more of that.

I’m GeePaw, and I’m done.

***

To read this article online with embedded hypertext links, go here:

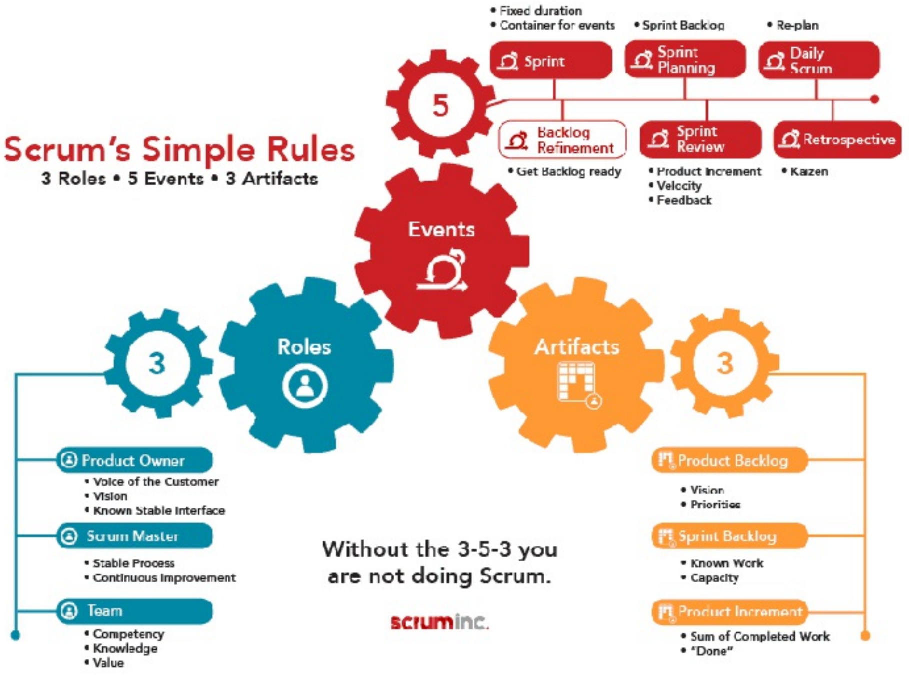
**http://geepawhill.org/coaching-like-people/**

# The 3-5-3 of Scrum

By Joe Justice

Posted on August 1, 2018

Scrum teams wishing to receive the return on investment associated with a rigorous Scrum should have an immediate method to check if they are implementing the same practices observed in the documented high performing teams. To support this, I encourage Scrum masters to check themselves against the Scrum guide and its 3 roles, 5 events, and 3 outputs. This shorthand reminder is used in my courses and I hope it presents some teams a quick-check tool to win the speed and happiness benefits observed in teams implementing the 11 components of Scrum.



### Importance of the 3 Roles

If a team is missing a Product Owner, they must ensure through some other method that the backlog is clearly communicated, rank ordered, and the stakeholders are made excited for what the team can actually produce.

If the Scrum Master role is missing, the team must ensure work is made visible and kaizen, or continuous improvement, is being driven through some other means.

If the Development Team is missing the team will have no method to create a working product with quality.

### A Business Argument for the 5 Events

Without Sprint Planning the team will need another method of formalizing all work is in direct support of the top priority and that a clear short-range plan exists to execute it.

Without the Daily Scrum the team will need another method to ensure high communication saturation (everyone on the team able to independently write down the same answer of the top priority goal, the plan to build it, and the constraints in which it must perform to be successful) or the team velocity may suffer as people implement their own varied understating of the mission. The team will also need another method to course correct, implement kaizen to accelerate and improve the solution, build team spirit, make impediments visible, and deal with disengaged or low-performance team members.

Product Backlog Refinement, or Grooming, and sometimes called Story Elaboration, is not an event in Scrum, however, the Scrum Guide requires the top of the backlog be ready for work. At Scrum Inc. we do require refinement, whether it is 15 minutes a day or up to 4 hours after lunch near the middle of the sprint, and without this the team will need some other method to progressively split large chunks of work into smaller chunks of work, clarify and pivot the rank ordered missions, add acceptance criteria, and bring the very top of the backlog to an immediately actionable and understood state.

Sprint Review results in feedback, excited stakeholders if done well, a pivoted and improved priority of what to do next, and a potentially shippable product increment. If the team has another method to accomplish these outputs they may transcend sprint review.

If the organization is interested in performance the groups will need some way to create a shared understanding of how the work is actually getting done and what is building or blocking it, then producing a kaizen (continuous improvement) to accelerate delivery, increase happiness, and drive up quality. The Sprint Retrospective is a short time commitment on a fixed recurring interval in which all of the above can be accomplished.

The US Navy 2018 report on high performing teams asserts that time boxing creates the near-term feedback loop required to release early and iteratively improve the result. The sprint in Scrum creates this and a rhythm, like the cadence of a professional sporting event, letting the teams exert and recover in strength-building anticipated cycles. Intense work without rhythm is an arrhythmia, and breaks down strength through chronic fatigue as the players are not able to anticipate the flow of exertion. Simply introducing a rhythm turns intense effort into strength and capability building for athletes and in my observation also in knowledge work and production.

Now that the 5 events have been, I believe, justified with business outcomes, I'd like to explore why the 3 outputs of Scrum are worth being rigorously checked in each Scrum team.

**The Value of the 3 Outputs**

Without a socialized long-range plan, investors cannot interact with the company or teams, the Product Owner cannot brief customers and stakeholders on what is coming soon in order to get them excited, and the teams are not able proactively build competencies or clarify needs in anticipation of upcoming work. The Product Backlog is the simplest form of product roadmap, simply an ordered list of missions and goals, and fundamental for a group to be compatible with investment.

A short-range plan with a clear true north, or compass, aligns the team to avoid building anything unnecessary and enable lean development. The Sprint Backlog with the sprint goal creates a stable mission for the length of the sprint so that team can swarm to deliver something coherent and valuable together.

A potentially shippable product increment allows the team to experiment quickly on a working concept, more clearly receive feedback from real users and stakeholders, solicit investment earlier in the process, and potentially ship early for the advantage of early capital.

I hope this clarifies why a team doing only 2 of the 11 parts of Scrum, commonly a Product Backlog and Daily Scrum, is unlikely to hit the fast delivery multiples for which Scrum infamous. I am not a fan of dogmatic process adherence, and any methods that create the values of the above 11 elements is highly likely to build fast teams and fast product. Even so, the 11 elements of Scrum are written so that they are self-reinforcing, the outputs of each is an input for another, making a robust and even viral system.

For these reasons, if your Scrum team is not in compliance with Scrum Guide, that is it is not executing 3-5-3 with the inputs and outputs listed above, your fist impediment to knock out in order to build speed is simply getting 3-5-3 working again. Scrum Inc. does firmly endorse transcending the 3-5-3 of Scrum, which we refer to as Ri (mastery) state Scrum, provided the values of the 3-5-3 are still accomplished in another way. Please think of 3-5-3 as the bootstrap, the minimum viable enabler that has a high success rate creating high performing teams if implemented with attention to the inputs and outputs of each element. A caution, it is, of course, possible to have a company full of 3-5-3 teams with no performance if the retrospective results in no kaizen, the sprint planning results in no clear short-range plan, the Scrum Master does not make work visible, etc. These zombie scrum teams can be caught and rescued by checking the input and output of the 3-5-3.

Keep up the awesome,
Joe

**Note:** Scrum was once even simpler, 3-3-3. Those were idyllic times. Esther Derby proposed to splint the Sprint Review in the Review of the Product (Sprint Review) and a separate event for Review of the Process (Sprint Retrospective). While both practices were explicitly listed as outcomes of the original single event, hosting 2 unique events has helped thousands of teams remember to inspect the product and the process. For more information please enjoy reading *Agile Retrospectives, Making Good Teams Great, by Diana Larsen and Esther Derby*.

***

To read this article online with embedded hypertext links, go here:

**https://www.scruminc.com/the-3-5-3-of-scrum/**

# About Joe Justice

**Joe** Justice is COO of Scrum Inc. Japan and works globally as an interim executive for agile organizations, bringing multinational companies twice the work in half the time. His teams have held 4 world records. He is a TEDx speaker, guest lecturer at both MIT and Oxford University in England, featured in Forbes 7 times to date including as owner of a "Company to watch" by Forbes Billionaire Club, cited in more than 5 business paperbacks and hardcovers, the subject of a Discovery Channel mini-documentary for his work creating the discipline Scrum@Hardware while working directly with the co-creator of Scrum, Dr. Jeff Sutherland.

Joe has worked with all of the top 3 military and defense contractors, autonomous and smart road technologies, ultra-lightweight structures, guest lectured at UC Berkeley, MIT, on behalf of Carnegie Melon, CU Denver, The University of Washington, spoken at Google, Microsoft, Zynga, Lockheed Martin, HP Labs, The Royal Bank of Canada, Pictet bank, and others. Joe's work has been featured in Forbes, Harvard Business Review, CNN Money, the Discovery Channel, and others.

# The War Zone - The Place in Every Organization Where Agile Meets Waterfall

By Erkan Kadir

Posted on December 4, 2018

You're leading an agile software delivery team in some capacity - maybe as a ScrumMaster, product leader or manager. Your team has worked hard to align with the Agile Manifesto and you're proud of their ability to build and ship software at regular intervals. You believe the business is better off as a result of your team's ability to learn and adapt to changing market conditions faster than before. Unfortunately, your work becomes more and more challenging as the differences between agile and the status quo become apparent. The more agile your team gets, the more you rub the rest of the organization the wrong way.

For example, you resist when the product team asks you to commit to the annual roadmap because you value responding to change over following a plan. The sales team is not happy with how quickly your team pivots because they use the annual roadmap to close deals. The architects feel excluded because you believe that the best architectures emerge from self-organizing teams. Stakeholders are frustrated that milestones keep changing. Executives believe that your team is unable to meet commitments and demand more predictability.

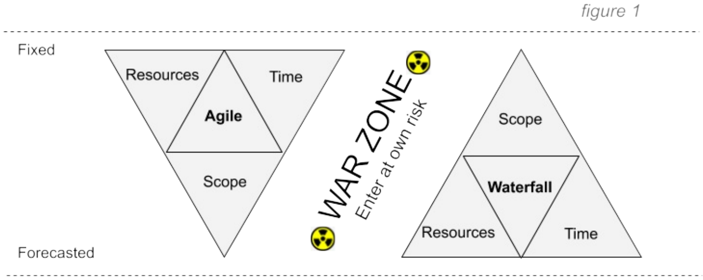
You realize that you're straddling the edge of two very different worlds, and to adapt, you begin to treat each world a little differently. Your vocabulary changes depending on your audience - "story points" and "spikes" for the engineers contend with "t-shirt sizes" and "project delivery dates" for the business team. You leverage agile principles with your team, then make long-term commitments to the business by padding your team's estimates. When the project plan needs to change, telling the business that you won't meet those commitments after all is a difficult conversation. Your team is operating under a completely different set of values, language, and expectations than everyone else. Misunderstandings, passive-aggressive behavior and full-on war abound.

If this sounds familiar, you might just be in the War Zone - the place in every organization where agile meets waterfall. When left unchecked, the War Zone grows more fierce and threatens your organization's success, but there are strategies that will help.

**A fundamental difference in approach**

Both agile and non-agile parts of the organization start new initiatives with good intentions and in alignment on a shared objective to be accomplished in a set timeline. Unfortunately, because of the unpredictable nature of new product development, neither agilists nor their counterparts have figured out how to hit delivery dates on time. The War Zone is fueled by a fundamental difference in how each side responds to these inevitable changes in schedule (*figure 1*).

figure 1



On one side we have the agilists who live by the values and principles laid out by the Agile Manifesto. They hold that their environment and technology are unpredictable, so they minimize planning activities and allow solutions to emerge through rapid experimentation. When it becomes apparent that a milestone cannot be met, agilists respond by solving as much of the underlying business problem as they can on the agreed upon date by deferring the least valuable features of the product. Agilist's build quality in and integrate work as they go, allowing them to delivery whatever value they have ready at any given time. They resist attempts to speed up by augmenting staff because they believe that adding people to a late software project only makes it later.

On the other side, we have those who are guided by traditional project management principles (typically the waterfall method). This side mitigates risk by controlling the environment and as many variables as possible. They value expertise in analyzing and planning for a single optimal solution. They find efficiency by working in batches where all planned product features are integrated and tested at the end of a project. In this all-or-nothing approach, dropping features to get back on track is typically not an option. So, the practitioner of waterfall will focus efforts to speed up on adding people to the project or pushing back the project delivery date.

**Every organization must navigate the War Zone**

The latest State of Agile Report from Version One has revealed that 97% of software companies globally practice agile methods while 84% are at or below a "still maturing" level. Stated another way, agile mixed with waterfall is the norm and we can expect to find a War Zone in almost every software company in the world.

The length of time an organization has been doing agile can be a good indicator of where the War Zone might appear. In grassroots adoptions, for example, the War Zone is almost guaranteed to appear early when every department other than engineering is unaware of agile. Even in mature agile organizations, the War Zone spontaneously flares up. For example, the War Zone appears whenever command and control leaders are appointed to work with agile teams, delivery of agile projects is managed through a PMO or the organization sells products which don't yet exist.
Highly aligned organizations may succeed in moving the War Zone out of their environments altogether, only to find that it now exists between them and their customers or suppliers when projects are expected to be delivered exactly as requested in a set timeline.

**We need each other**

No approach is better than the other. Agile and waterfall are each fit to task to solve certain problems. Agile practices are fit to solve problems with a high degree of unpredictability and the reverse is true for waterfall. Both agile thinking and the waterfall approach are necessary for organizations, but there's often a lack of understanding of when and where to deploy each method. Misunderstanding creates a lack of appreciation for what each method brings to an organization. Winning the war is not about total domination of one side or the other - both agile and waterfall are here to stay. Organizations who learn how to balance both approaches while developing strong relationships across the boundaries of each approach can minimize or even eliminate the War Zone.

**Make software not war**

Here's a six-step framework you can use to regain control over your War Zone.

1. Assess

- Where in your organization does the War Zone exist - vertically between layers of the organization or horizontally between departments? How serious has the conflict become? Frameworks like Speed Lee's 5 levels of conflict can help determine the severity of the War Zone.

2. Assemble

- Bring together the people and perspectives who camp on both sides of the War Zone in order to negotiate a peace treaty.

3. Normalize

- Let people on both sides know that the War Zone exists in every organization and that conflict is normal. The conflict they experience arises from a diversity of skill sets which are all valuable and necessary for a strong organization.

4. Educate

- Your organization and its people are resilient. Once they are aware of the War Zone, they will naturally choose a more skillful way to interact.
- Educate about the differences between the two approaches.
- Have each side openly discuss their values, beliefs, and expectations.
- Ask people to find what's valuable on the other side and what skills they might like a little more of on their side.
- Demonstrate the cost of the War Zone. How much better could they be if they focused that effort outward on competitors, instead of inwards on each other?

5. Negotiate peace

- Create a plan for how the two sides will communicate that is written down and posted in a highly visible place as a reminder. It should include:
    - A common set of words that will be used by both sides.
    - When and how often status will be communicated.
    - How they will show respect for the approach of the other side.
    - An agreed-upon strategy for responding to inevitable changes to schedule.

6. Make it stick

- Leverage support structures like posters or this sticker for people to remember how they want to be with each other.

Where does waterfall meet agile in your organization? How severe is the conflict? What is the impact to your business? What will you do about it?

***

To read this article online with embedded hypertext links, go here:

**https://www.superheroes.academy/blog/2018/12/2/the-war-zone-the-place-in-every-organization-where-agile-meets-waterfall**

# About Erkan Kadir

**Erkan** Kadir is a co-founder of the Superheroes Academy, a coaching organization that's set out to level up the skills of agile practitioners worldwide. He is a Scrum Alliance Certified Enterprise Coach (CEC), Certified Team Coach (CTC), Certified Organization and Relationship Systems Coach (ORSCC), and an IC-Agile Certified Coach (IC-ACC). Erkan leverages 15 years of experience developing software, managing, and leading organizations to help grow a diverse range of clients from teams and organizations to family systems.

Write him at kadire@gmail.com or https://www.superheroes.academy/

***

# Sizing a spike?

By Vamsi Krishna Kakkireni

Posted on July 23, 2018

Agile world is vertically split on the point whether to size a spike story? Well, there is no absolute right or wrong answer, but this is to help you in understanding factors to size it or not. And let us also get down to the nitty-gritty of what it is and what are the various observations and supporting statements around it.

**Spike story** – A story aimed at resolving something or answering some questions or gathering information, rather than producing a releasable product. Typically, a spike is used for activities such as research, design, investigation, exploration, and prototyping.

**Reasons for sizing a spike story**

1. Give credit to the team for the work it is doing. As finishing a spike also needs some effort.
2. We are just giving a rough estimate and that is used only for release planning purpose.
3. With spike, team has increased their knowledge. And next time they use this knowledge and relatively size it less for similar stories.
4. Efforts on spike should be time-boxed as we cannot go on for indefinite period. And sizing a spike is an art. We need to brainstorm on the unknowns and risk.
5. We are sizing only for a capacity planning purpose and later we remove it from velocity.

**Reasons for NOT sizing a spike story**

1. Don’t want to inflate velocity.
2. It does not add any direct value to customer and all it gives is just a potential solution for other piece of work.
3. We are not sure on the unknowns.
4. We cannot relatively size it as we cannot compare with other spikes.

**Some pointers to help you**

- Consider an example, “if technology xyz can scale a million users.” In this case, it is better to **size the spike** as this will be independent of anything in product backlog. Here we are not getting double credit for the same piece of work. I recommend sizing only for this type of cases.

- Consider a scenario where a team is creating a spike story for a PoC in which they are trying out different options, and if team chooses one of those options as a solution. And if you decide to **size the spike story then do NOT size the other story** provided it is not taking too much of effort. (as there is no unknown, no risk, no complexity and a very little amount of work to complete the actual story) – This is to avoid double credit.

- Suppose if you have a spike to “design the ABC component” and then a product backlog item to “develop an ABC component” then in this case **do NOT size that spike** since that would really be a double count of points earned for the work.

- Every user story has a bit of risk, complexity unknown/uncertainty. And that doesn’t mean we create a spike for every user story. So, brainstorm if we really need a separate spike. Be true to yourself.

- Get everyone on to the same page including stakeholders about the purpose of a spike and communicate it upfront (if we are sizing or not) so that the expectation and commitment is clear. It is recommended Not to play with it like sizing for a few spikes and not sizing a few. Consistency is the key here.

- Regardless of sizing, spikes are time-boxed. So, it is better to have a mid-sprint review and discuss if we can arrive at a solution in the given time-box i.e. within the sprint. It is also not wrong if it spills over to next sprint.

(depending on the complexity and unknowns)

Either way it is important to acknowledge the learning from a spike. Re-emphasizing that there is no right or wrong answer. But it is much needed to keep a tab if you are claiming double credit.

***

To read this article online with embedded hypertext links, go here:

**https://agileenthusiast.com/2018/07/23/sizing-a-spike/**

# About Vamsi Krishna Kakkireni

**Vamsi** is an agile enthusiast with overall 12+ years of experience in IT. He is a blogger and also played multiple roles - Scrum Master, Project Lead, Developer, Tester & Business Analyst. Vamsi is certified in PSM1, DevOps (CDP), KMP1, SAFe SA, SSGB & Pega CSA. He is passionate in Scrum Master role and enjoys connecting with agilists by participating, attending conferences.

His goal is to help organizations build awesome products by enabling agility.

He recently spoke at Open Space Session at Regional Scrum Gathering India on '*Agile for Customer Delight, It's neither B2B nor C2C but it's H2H*'.

You can connect him at blog (https://agileenthusiast.com) or Linkedin (https://www.linkedin.com/in/vamsikrishnakakkireni/)

# How to Build Your Own “Spotify Model”

By Juriaan Kamer

Posted on February 9, 2018

It is early 2011 and you are the CTO of Spotify. You’re staring out of the window of a coffee bar in a snowy and dark Stockholm. It has been an amazing year. Your company is acquiring customers faster than ever and you’re launching in more and more countries. However, Google and Apple are catching up. It’s not a question if they will launch their own music streaming services, but *when*. To survive that eventuality Spotify must become a truly global player.

You have to do this while still figuring out how the music streaming business actually works. What do customers really want? What will they pay for? What do you need to do to convince someone to stop buying CDs or MP3s and instead be willing to pay for a monthly streaming subscription?

“We need to innovate, experiment and learn faster than the competition.”

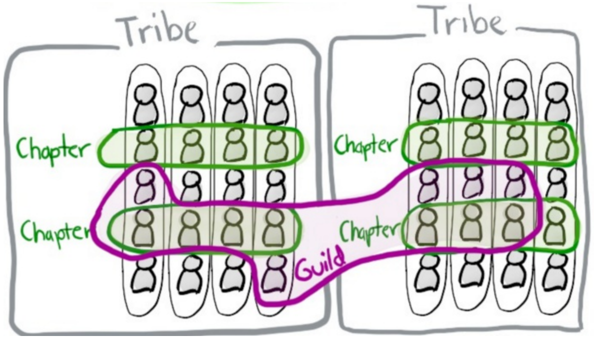
To do all this on a global scale you have to grow your engineering team. It has been a huge challenge to grow your team from 10 to 100 in the last year. Some people predict you might even need to attract another 1,000 engineers to pull this off. You feel overwhelmed. How can you attract the right talent with the right mindset across the entire globe?

Managing a team of 100 is already complicated. But if you grow even further, how do you stay agile? How can you keep the start-up culture that has brought success so far and prevent becoming a lumbering bureaucracy?

# Spotify’s organization design

Another two years have passed and the company now has 15 million customers. The engineering team has tripled in size to 300 people. A challenge that has been top of mind for a while now is, *“With 30 teams, how do we make sure we build a castle that makes sense to the customer, instead of a pile of 30 bricks that nobody likes?”*

The teams have started to experiment with a scaling model that uses Squads, Chapters, Guilds and Tribes who aim to implement ‘minimum viable bureaucracy’ and balance *high autonomy with high alignment*.



Source: Spotify's engineering culture

This structure is just one piece of the puzzle, though. Through a few workshops, the agile coaches came up with a set of organizational design principles with the *autonomous team* as the core concept. This extension of the agile manifesto is called "Agile à la Spotify", and has been printed on the walls across the office:

- **Continuous improvement:** At Spotify, part of my work is to look for ways to continuously improve, both personally, and in the wider organisation.
- **Iterative development:** Spotify believes in short learning cycles, so that we can validate our assumptions as quickly as possible.
- **Simplicity:** Scaling what we do is key to Spotify's success. Simplicity should be your guidance during scaling. This is as true for our technical solutions, as for our methods of working and organising the organisation.
- **Trust:** At Spotify we trust our people and teams to make informed decisions about the way they work and what they work on.
- **Servant leadership:** At Spotify managers are focused on coaching, mentorship, and solving impediments rather than telling people what to do.

# Full autonomy is a trade-off

Fast forward to early 2018. Spotify has 140 million users in 61 countries. It has announced an IPO at a $20 billion valuation. Engineering and R&D is now 180 teams and 1,800 people. In total, Spotify employs over 3,500 people: the organization is no longer a startup, it's a global enterprise.

A lot has gone really well. The culture is known for a high level of empowerment and trust, a focus on personal development, and is known for its sense of purpose. Teams are fully empowered to fulfill their mission and have the freedom to act independently. But as Joakim Sundén pointed out, it is far from an agile nirvana.

"What is the best thing about working at Spotify? What is the most challenging thing about working at Spotify? The answer for both questions is the same: Autonomy." — Joakim Sundén

Managing 180 autonomous teams can feel like herding cats, especially when it comes to projects that span across teams.

Some recent examples: implementing SOX compliance to enable the IPO, keeping up with new privacy laws and moving all infrastructure into the Google Cloud. Also, the lack of focus on architecture and technical standards has made it challenging to scale the platform to support its growing user base.

Implementing these initiatives top-down wouldn't work in Spotify's culture. A team can simply say "I don't want to do it" and would need to be seduced to give these priority.

Over the years, the lack of central planning and standardization has enabled hyper-speed, hyper-growth and hyper-innovation. But it has made certain things a lot harder that are easy for more traditional organizations.

Time will tell how Spotify will continue to evolve. It's a challenging balancing act between doing the *right things*, doing the *things righ*t and doing *things fast*.

# Don't copy the model

As agile organization designers, we've been following Spotify closely. Over the years, we've visited their offices several times. It's an awesome company and there is much to learn from them. We love how the engineering culture videos have inspired thousands of people to start upgrading their organizations.

However, if you're considering implementing the 'Spotify model', please think again. Is *you*r organization building a music player? Is *your* organization still trying figure out its business model? Is *your* organization facing hyper-growth? Is "move fast and break things" applicable to *your* product?" Maybe. But, probably not.

When people copy the 'Spotify model', it often happens through a top-down directive, without taking a close look at what kind of culture and leadership is needed to make it work. Often the existing hierarchy is changed into a new, static matrix blueprint (even if the matrix is labeled with Squads, Chapters, and Tribes), instead of growing a culture of continuous participatory change. This will inevitably make things worse. Even people who work at Spotify recommend to not copy their model.

Don't get us wrong: in order to enable agility in an organization, we do recommend that you move away from top-down management and focus on empowering capable teams. But to copy a pre-existing model and believe that your problems will also be solved, is short-sighted and naive.

“The only Spotify way of working that actually works is turning on the Spotify volume really loud and dance.” — Erwin Verweij

# Evolve your own organizational model instead

> Just as startups are focused on finding product-market fit, we believe you should start on a journey to **find your organization-context fit**. Spotify has been able to do both.

We love this quote that is at the heart of what we believe:

> “Stop trying to borrow wisdom and think for yourself. Face your difficulties and think and think and think and solve your problems yourself. Suffering and difficulties provide opportunities to become better. Success is never giving up.” — Taiichi Ohno

So what can you do if you want to gain agility, speed and innovation? Where to begin?

First, ask yourself, is there a clear picture of **what issues you’re trying to solve** with a new organizational model? If possible, find a few measurable indicators of what needs to be improved.

Involve not only your leadership team, but also a wide variety of people in the organization to gather ideas and **co-create a picture of the desired future state**. A good question to ask is: *what is holding you back from doing the best work of your lives?*

Don’t forget to **appreciate what is going really well** and decide what you definitely want to keep.

Take inspiration from a wide variety of future of work practices and companies. **Look at different models of self-organization** that fit different scale and risk contexts. Look beyond Spotify and even look beyond agile to gain org-wide agility.

Figure out what the **main capabilities** are you need to upgrade and where in the organization they are rooted. The OS Canvas is a useful tool for this exercise.

**Design and start a number of pilots** that help you try out new ways of working and allows you to quickly learn if it fits your specific context. Expand the pilots that show promise. End the ones that don’t produce the effect you’re looking for. Build your organization’s ability to constantly try new behaviors and learning from those experiments.

At the end of the day, use the Spotify model as an inspiration for what’s possible when you spend time and attention developing your own operating system — not as a model for what your own system may end up looking like. Design, test, and evolve your own model as inclusively as possible. Don’t do a big-bang change towards a new static target operating model, but instead build the muscle for continuous participatory change.

Don’t “do the Spotify model” — do your model.

---

**Co-authored** by a friend of The Ready: Roy Gielen (Agile enabler, trainer & coach at Ctree)

---

Did you know that The Ready founder, Aaron Dignan, recently finished a new book all about how organizations can change the way they operate? If you’re interested in the future of work movement, the work we do at The Ready, or simply how to make your own organization better, then you should definitely order your copy of Brave New Work today.

***

To read this article online with embedded hypertext links, go here:

**https://medium.com/the-ready/how-to-build-your-own-spotify-model-dce98025d32f**

# About Juriaan Kamer

**Jurriaan** Kamer is an organization designer, transformation coach, author and speaker. He is a Partner at The Ready, an organization design and transformation agency focused on the future of work. Through The Ready, he helps leaders free their organizations from the rules, habits, and mindsets that cause delays, making work faster, more agile, more human, and more fun. He regularly gives presentations and workshops about his experiences and practical examples to inspire and instill change.

His recently released book *Formula X* (www.formula-x.co) *"how to reach extreme acceleration in your organization"* is a business novel about speed, leadership and organizational change, inspired by Jurriaan's passion of Formula 1 racing.

Feel free to contact Jurriaan or subscribe to his newsletter via www.jurriaankamer.com. Or follow him on LinkedIn or Twitter (@kajurria) to receive a stream of interesting articles.

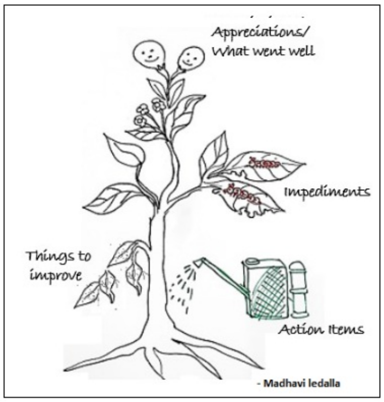
# Garden Your Thoughts

By Madhavi Ledalla

Posted on April 9, 2018

## Situation

As I was watering the plants in my garden one lazy Sunday morning, I found a few insects perched on some leaves. Initially, I ignored them, thinking they would disappear on their own. However, within a week, I saw that the whole plant was eaten up and it was almost about to perish.



Our work also may indicate similar threats early in their making. Often we tend to ignore early signs of problems because we believe they don't demand our urgency. In collaborative working environments, any small issue left unattended to may end up slowing down the entire value chain.

This should definitely remind us of a few scenarios we've encountered in our daily work. Often, we see impediments coming our way, but we ignore them because we believe they don't demand our urgency. However, these things eventually pose a major risk for the entire project. What could have been avoided will become an impossible problem. This situation is particularly relevant to agile teams. Because collaboration is crucial for the health of agile lifecycles, any small issue left unattended ends up slowing down the entire value chain.

## Name and Description- Garden Your Thoughts

The next time I was conducting a retrospective with teams, I drew a picture of a plant on the board. "Look," I said, "this plant is our project, and I would like us to do the following things."

Think of all the bottlenecks that, if not addressed, would eat us up like a hungry worm devours leaves. Identify best practices that really worked well for us, and continue to work on them, just as we would care for flowers. Realize collectively what it is that we want to try different so that we stay on the right track, just like how we would nourish the plant with fertilizer and pesticides, the necessary action items. Lastly, identify potential areas of improvement, as we would tend to the drooping leaves on a plant.

## Instructions:

Draw a plant on the whiteboard and ask your team to think of it as their project. Instruct them to:
1. Think of all the bottlenecks which, if not thought of, would damage the project like a hungry worm on leaves.
2. Identify best practices that worked well so far and continue to work on those just as we would care for the plants.

3. Realize collectively what it is that we want to try differently, the action items to stay on the right track, to secure the project(a plant with necessary fertilizers and pesticides)
4. Identify our areas of improvement, just how we would tend to the drooping leaves on a plant. This would help draw up good data points for discussion.

## Material

Large white board or a large piece of poster paper. Three colors of sticky notes and markers.

***

To read this article online with embedded hypertext links, go here:

**https://lmadhavi.wordpress.com/2018/04/09/garden-your-thoughts/**

# About Madhavi Ledalla

**Madhavi** Ledalla is an Agile Coach based out of Hyderabad, India. Madhavi is a proponent of visualization and a firm believer of the saying- “*A picture is worth a thousand words*”. This led her to research, create and experiment with collaboration frameworks, an artful coaching technique of creating lightweight structures for engaging teams to help them explore and discover their universe.

She has a keen interest in delivering customized workshops for teams and leadership. Her practices include training and coaching in varied agile frameworks. She has played multiple roles of a Scrum Master, Agile Coach, Project Manager, and a Technologist. She brings in perspectives of both traditional and agile methods. She is a Speaker, Reviewer & Organizer in Regional and Global Agile conferences.

She occasionally blogs @ https://lmadhavi.wordpress.com/

# What the heck is self awareness and why should you care?

By Kate Leto

Posted on June 4, 2018

> The biggest opportunity for improvement — in business, at home, and in life — is awareness.
>
> — Alan Mulally, former CEO of Ford Motor Company

In 2006, Alan Mulally took over as president and CEO of a struggling Ford Motor Company. Ford's glory days as one of America's top automotive companies were a distant past, and the company was staring at $17 billion losses. While other notable industry leaders from Daimler AG and Renault/Nissan had turned down the offer to take the top role, Mulally jumped on board.

In just over five years, Ford was reporting $20 billion in profits and was the only automotive company in the US to avoid a government bail-out.

What was the secret sauce to Mulally's success? You could point to his success at negotiating new contracts with labour unions to cut costs, increases in organisational efficiencies, or his introduction of a strong vision to enhance stakeholder value.

But, when asked about the greatest moment of the turnaround, Mulally says it was when his executive team started being comfortable enough with him to tell him the truth. Once this team of senior executives realised that they could count on him and each other to get through their seemingly insurmountable challenges, the turnaround really started.

In her book, *Insight: The Power of Self Awareness in a Self-Deluded World*, organisational psychologist Tasha Eurich investigates Mulally's drive to create a culture based on his own work towards self-awareness; something he attributes to an earlier experience when a favourite member of his team at Boeing quit suddenly. That moment struck him like thunderbolt, igniting a passion to learn more about himself and his impact on others.

# What is Self-Awareness Anyway?

Good leaders, like Mulally, may make the work look easy, but they've done a lot of work on themselves to get there. To get an understanding of what type of work we're talking about, let's start with a basic definition of self-awareness.

Eurich defines self-awareness as:

*"The ability to see ourselves clearly — to understand who we are, how others see us, and how we fit into the world around us."*

What I like about Eurich's definition is that it pushes the boundaries on the more traditional thinking of self-awareness, which often focuses on an inward study of self — how we think, feel and act — with less focus on how our actions impact others.

Eurich's approach is more multi-dimensional and incorporates inputs from the outside world — be it from your partner, team or the organisation you work within — into a more rounded view of self. The world outside of us becomes stimuli for self-awareness.

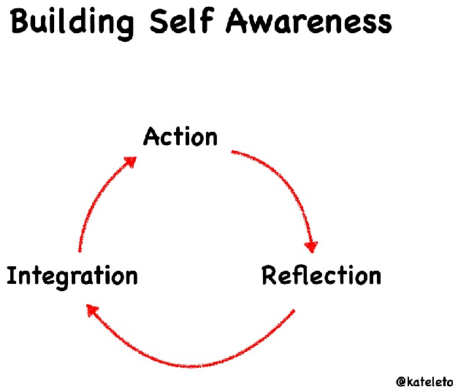
# The Business Case for Self-Awareness

The theologian and philosopher St. Augustine said:

*“People travel to wonder at the height of mountains, at the huge waves of the sea, at the long courses of rivers, at the vast compass of the ocean, at the circular motion of the stars; and they pass by themselves without wondering.”*

While for some, the concept of self-awareness may be way too “touchy feely” to consider (or perhaps even read about), for those who do take the time to focus on growing self awareness, the benefits can be life changing — for you and for those around you.

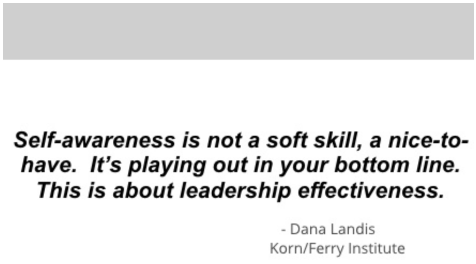
I can speak from experience. As someone who has been a student of my own self-awareness over many years, I can say that the experience may feel like you’re on a never-ending road to nowhere at times. However, the outcome so far has been beyond worthwhile both personally and professionally. While I still have a lot to learn about myself and how my actions affect others, I’m happier, healthier, and basically just feel better in my own skin. It’s an ongoing process of action, reflection, and integration that I value and now try to work with others to engage with. (Yes, remarkably similar to the build-measure-learn approach we all work with daily!)



The data says that I’m not alone in that feeling. Research shows that with a stronger sense of self-awareness, we are more aware of our strengths and weaknesses, helping us to make better decisions, be more creative, build better relationships, be more empathetic, communicate better, and often get more promotions at work. It’s no wonder that Eurich calls self-awareness the “meta skill of our time”.

For those more interested in the bottom line, check out “*A Better Return on Self-Awareness*” by Korn Ferry Institute analysts David Zes and Dana Landis. This whitepaper shares data from 486 publicly-traded companies over a 30-month period from July 2010 to January 2013 that shows a link between self-awareness and business performance. According to Zes and Landis:

- Public companies with a higher rate of return (ROR) also employ professionals who exhibit higher levels of self-awareness.
- Poor-performing companies’ employees were 79% more likely to have low overall self-awareness than those at firms with robust ROR.



***Self-awareness is not a soft skill, a nice-to-have.  It's playing out in your bottom line. This is about leadership effectiveness.***

- Dana Landis
Korn/Ferry Institute

With that type of impact on a company's bottom line, it's not surprising that the more immediate impact can be felt on your team. According to Eurich, employees who lack self-awareness:

- bring down team performance, reducing decision quality by an average of 36%
- hurt coordination by 46%
- experience 30% more conflict

# Think You're Self-Aware? Take Another Look

According to Eunich's research, 95% of people think they're self-aware, but the real number is closer to 10% to 15%. In another study of 17,000 people worldwide by the Hay Group Research, 19% of women executives and 4% men interviewed displayed self-awareness. No matter how you slice it, we all could use some focus on building that self-awareness muscle.

Thankfully, as with all of the competencies of EQ, you can do just that. But, just as I discussed in the post on conflict resolution it's going to take time and commitment.

Here's three ways that you can start:-

# 1. Understand Your Core Values

Unless we understand our own values we aren't going to be able to make decisions in our best interests.

As part of my coaching practice, I often work with clients on values to build self-awareness around who that person is, what's important to them, what's immovable, and how they are living that value every day.

Recently I worked with a product manager who was in a very dry, technical position in an aggressive organisational culture that he was very unhappy in. He had tried just about everything to make the job and the company right for him — tried to improve his stakeholder management skills, experimented with team rituals and ceremonies and much more — but nothing was working.

In one session we did a values exercise to better identify his key values (five or six values that resonate with fundamental, core beliefs). He selected creativity as a key value, but was working in a very dry, technical position in an aggressive organisation that restricted creativity.

He realised that the tension that existed between his need for creativity and his company's culture was making him stressed, frustrated, and just plain fed up. As a result, he became more confident and clear about what he needed from a working culture and has since moved on to a new product role that embraces creativity.

# 2. Identify Your Blind Spots Through Continual Feed-

# back

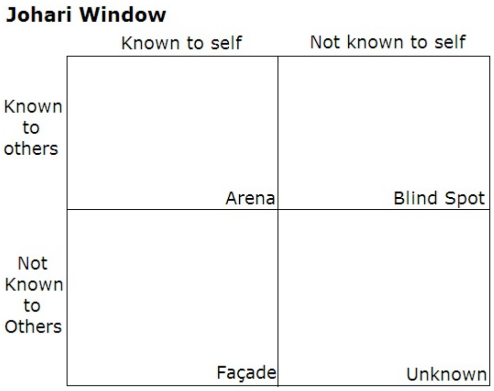
According to Robert Bruce Shaw, author of *Leadership Blindspots: How Successful Leaders Identify and Overcome the Weaknesses That Matter*, a blind spot is an unrecognised weakness or threat that can undermine your success. Uncovering and addressing them is a huge opportunity for self-awareness.

One of the best ways to beat your blind spots is to ask for honest feedback. I know, it's something we talk about all the time — the importance of feedback, how to give and receive it, the fact that (unlike Christmas presents) it shouldn't be given just once a year at performance reviews. I'm going to go on about feedback again, but encourage you to think of it with a slightly different model.

Marshall Goldsmith is an executive coach and author who says that we tend to accept feedback that is aligned with our view of self, and reject feedback that doesn't match our view of reality. In his book, *What Got You Here Won't Get You There* he says constructive feedback is based on four components:

1. Ask the right people
2. Ask the right questions
3. Interpret the answer properly
4. Accept the responses as accurate

A tool that I use to facilitate constructive feedback in coaching is the Johari window. A concept created in 1955 by American psychologists Joseph Luft and Harrington Ingham (aka Johari), this simple tool helps others see who you think you are and how you see yourself.



Source: Wikipedia

Try the Johari Window with your team:

1. Pick five or six adjectives from a list that you feel describes your own personality
2. Select three to five people you work with to also pick five or six adjectives to represent you
3. Once you receive all the feedback, place it in the appropriate area of the "window" (you can read more about placement in the windows here)

It's interesting to repeat this exercise at different phases of your team's development — be it forming, storming, norming, performing.

# 3. Have Regular Check-ins With Yourself

We all have crazy, busy days and can easily lose track of ourselves in the ongoing cycle of commute, work, home, repeat.

In my own efforts to break this cycle, and grow my own self-awareness, I’ve found that simply taking a few minutes a day to check in with myself can be a huge benefit. Be it mid commute on the tube or bus, in a conference room before a meeting, walking down the street, drinking a cup of coffee, or even washing dishes — I find an opportunity to clear my mind, just focus on my breath and check in on how I’m feeling, what I’m worried about, or maybe how I’d like to handle an upcoming meeting.

In some ways, it’s often a mini-moving meditation that helps me retain my focus and get a bit more grounded.

Find what works for you. For some, it might be running, going to a yoga class, or simply walking around the block. For others that might be journaling, going to art museums, or meditation. Find what’s right for you.

Because at the end of the day it all comes from you. You do the hard work. But that intention can take you — your family, team, and even organisation — to a far better place.

***

To read this article online with embedded hypertext links, go here:

**https://medium.com/the-ready/what-the-heck-is-self-awareness-and-why-should-you-care-4843ab6a7cfa**

# About Kate Leto

**Kate** is an advisor, coach and consultant with more than 20 years of product management, marketing and organisational design experience.

Since 2011 she's been an independent strategic product consultant, leading organisational design and transformation initiatives across finance, government, travel, e-commerce, and public works sectors. She is an executive and innovation coach, working with individuals, teams and organisations to achieve their evolving aspirations.

In early 2017, she began writing about Product EQ - an effort to cultivate emotional intelligence in product management and leadership. Read more on the topic here: https://medium.com/product-eq

Kate is an international speaker and facilitator in the product, innovation and transformation spaces.

# How to Cross-Skill and Grow T-shaped Team Members

By Mark Levison

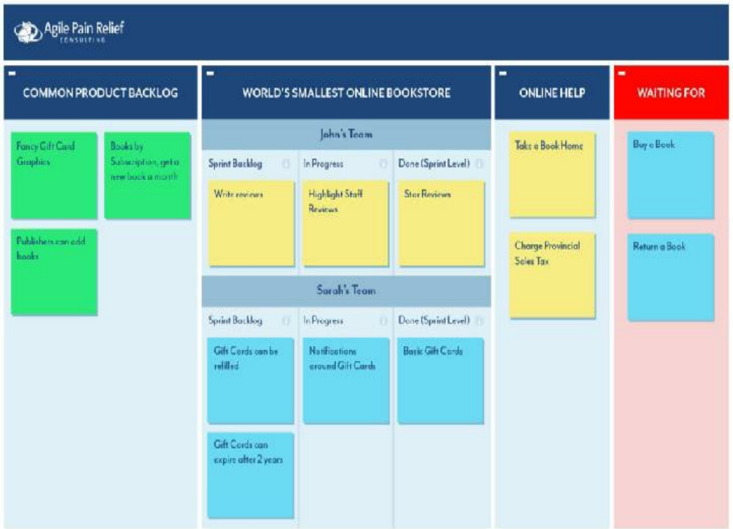
Posted on June 26, 2018

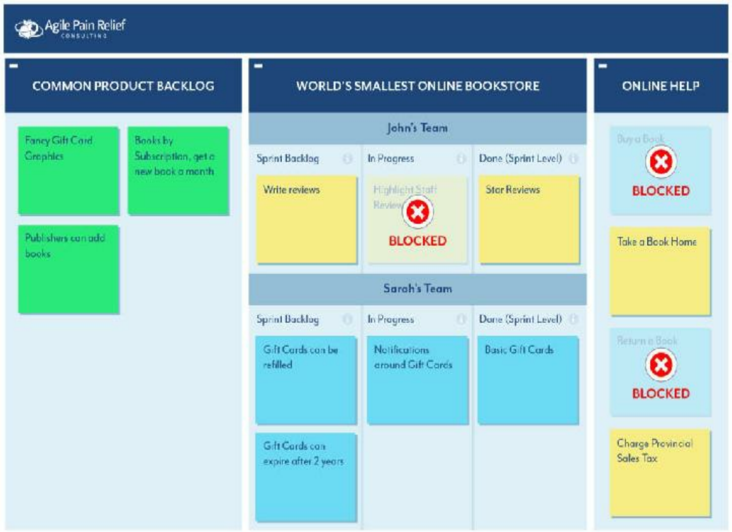
As we discussed in "Specialists Are Overrated," developing cross-skills and "T-Shaped" people in a team has many benefits – for the team/organization itself, the customer, and the individual. That's all fine and good to say, but how do you figure out where to start?

There are two major ways to discover opportunities for cross-skilling:

1. Kanban/Scrum Team Board
2. Skills Matrix

## Kanban/Scrum Team Board





The Team Board is a rich source of information as to where missing skills exist among a team's members. If you're unfamiliar with how a Team Board works or what it looks like, I posted an example of a Kanban Board previously on this blog. Walk along the board every few days, look to see which column work items use up the most time. The board should also help you spot any item that gets blocked or is waiting for outside work. Every time an item is blocked because of an external dependency, record the reason. Over a few months, it will become clear which external dependencies impede work getting done the most. Any of these areas discovered are now opportunities for the team to cross-skill into. Once there is enough evidence to start (perhaps after 3-4 Sprints of collecting), share the data in a retrospective and ask the team two key questions:

## Theory of Constraints

Demonstrates that in any system where there is a bottleneck, the rest of the system should be subordinated to the bottleneck until the bottleneck is cleared. Goldratt's work is, in part, the basis for Kanban and also shows why, in a constrained situation, moving a Developer or Writer from their primary work to the constraint (e.g. Quality Assurance or Editing) is so effective.

- "What area would we like to put energy into learning?"
- "Who has the interest and energy to learn about these areas in the next few Sprints?"

If they do this well, the Team should rarely find itself stuck waiting for an external dependency because it will have grown the skills to handle many previously external dependencies inside the team. In addition, many bottlenecks, like a lack of Quality Assurance or editing skills, will have been resolved when those activities become a constraint [1] team easily adapts to changing needs.

Using the board as a source of data will also help reveal issues that the team has faced in the recent past. However, it only works as long as the future work looks similar to previous work. So it's a necessary tool, but one with limitations.

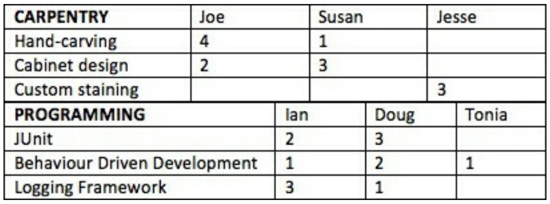
## Skills Matrix

While the Kanban/Scrum Board helps the team understand current and recent challenges, it doesn't do anything to address where the team will likely need new skills going forward, nor where team members would like to grow in their skills.

A Skills Matrix is a self-reporting system where team members provide their own estimate of their skill in a specific area. To create a skills matrix, get the team to set aside a couple of hours and run a workshop with the following steps:

1. On a large piece of paper, write down all of the skills you personally have that are relevant to the work. Include your superhero skill (or anything else that will provoke a smile).
2. Pass your page to the next person. They review your list and add any skills for you that they feel you missed.
3. Repeat until people aren't adding any new skills to the lists. Typically this will happen after about three people.
4. Compile all of the personal lists into a single long list that everyone can see. In skill areas that are of greater value or importance to the team, go into greater detail. For example, inside of Java Development, we might write down specific libraries or tools that team members use.
5. Team members' names are written down on one axis, skill areas on the other.
6. Team members self-rate their skill level in each area. Any scale can be used; for example, mine typically goes from: Blank – don't want to learn about this and 0 – knows nothing but open to learning, through 2 – can complete small tasks unaided by an adult; up to 4 – expert in the area and others can learn from me.
7. If we compute the average for each skill area, we rapidly get a picture of where the team is strong and experienced, and where it's weaker.

### Sample Skills Matrix



| CARPENTRY | Joe | Susan | Jesse |
|---|---|---|---|
| Hand-carving | 4 | 1 | |
| Cabinet design | 2 | 3 | |
| Custom staining | | | 3 |
| **PROGRAMMING** | **Ian** | **Doug** | **Tonia** |
| JUnit | 2 | 3 | |
| Behaviour Driven Development | 1 | 2 | 1 |
| Logging Framework | 3 | 1 | |

I like to create a Skills Matrix whenever I start working with a new team. If we haven't started work yet, then I do it at the same time as we create the initial Definition of "Done", before the first Sprint. Once the Skills Matrix has been created, I recommend teams revisit it every few retrospectives to answer two questions:

- "Where have we learned new skills that would cause us to update our self ratings?"; and,
- "Where would we next like to put our learning energy?"

An important thing to remember with the Skills Matrix is that it can only be used for the team by the team. If it is used outside of the team, members will game their numbers to look good, and that destroys the only value of the tool. All too often, I hear of organizations where Skills Matrices are a function of Human Resources and the information is used to poach team members for other projects —this approach is 180 degrees from the Agile use of the tool. I've also seen it misused by management to pressure team members or in the Performance Review process. If this happens, all value will be destroyed. It is a tool for the team to understand itself.

In an organization that is sufficiently mature that it won't get misused, I like to put the Skills Matrix up on the Team Room wall as a reminder of the importance of ongoing learning.

## Growing Knowledge

Skills Matrices and Scrum/Kanban Team Boards will only get you to the point where you know there is a need for growth, and where. Using that information in Sprint Planning helps remind you to set aside time for skills growth, but we still haven't grown a skill yet.

New Skill
Core Skill (Expert)
New Skill

T-Shaped people don't grow on trees.

Cross-skilling is important in every industry, and in each one there will obviously be different approaches to grow these skills. If you are in an IT environment, here are tools that you can use right now to help you grow skills related to software development and design:

**Pair Programming** – two people work with one computer to produce one work item. Its primary benefit is to reduce the complexity of the system being built, thereby reducing defects. However, another benefit is the rapid spread of knowledge it engenders.  *"Pair programing [sic] rocks. Taught other developer something last week, he taught to his partner, and I just heard it spread to a 4th team member"*[2]. This technique can just as easily be used in non-technical/non-software-related work – Pair Writing[3] is just one example, Pair Learning[4] another.

**Coding Dojo**[5] – a safe place to practice, where the team can learn how to learn. A group of team members get together to practice with a programming challenge. The challenge can be any simple programming problem that gives people the opportunity to practice a desired skill (e.g. Test Driven Development, Behaviour Driven Development). Good sources of problems: http://codingdojo.org/kata/, http://www.codekatas.org/ and http://codekata.com/.

**Learning Time** – related to a Coding Dojo, this pattern formalizes the importance and value of learning. Many organizations expect their people to do professional reading on their own time, however, that sends a signal that work is more important than family or socializing with friends. Consider instead having the team set aside a block of a couple of hours to work on learning. There is a preference for group work, for example a Coding Dojo over personal coding practice; a Book Discussion workshop over individual reading time. Sometimes, instead of reading, it can be a presentation by someone with a different background or skill area for the rest of the team. Learning Time works best when it's interactive and there is limited formal presentation.

When a team member needs time to read a book, consider asking them to write a short summary of each chapter they've read, as a way of sharing their learning back with the team.

**Community of Practice** – a way of spreading knowledge and skills outside the team. Pick an area that will be of interest to many people outside of one team (e.g. Testing in an Agile world; Use of Scrum Outside of Software). Establish a meeting frequency (e.g. every 4-6 weeks) and time. For the first couple of times, seed the agenda, polling the group to find out what their needs are. By the third event, ideally we want the audience taking control of the community of practice themselves with the goal to make it a self-sustaining entity.

## A Final Note

Even with all the cross-skilling in the world, some tasks will always require experts. But cross-skilling still helps in those instances, because the non-experts can help by relieving the expert of simpler tasks so they can tackle the bottleneck.

In the short term, perceived productivity will go down, because learning and growth always slow us down initially. But in the medium term, it improves quality, which increases throughput going forward.

The most important point to remember is that cross-skilling and skills matrixes only work when people don't feel coerced. We can create the space where people choose to grow and cross-skill, but we can't force them to learn.

**If you want to go faster, stop focusing on the speed. Slow down. Take the time to learn.**

**Focus on the work item and not the worker.**

Other References:

Jason Yip – "Why T-Shaped People"
https://medium.com/@jchyip/why-t-shaped-people-e8706198e437

Andrew Fuqua on the "Theory of Constraints"

[1] See the Theory of Constraints: https://en.wikipedia.org/wiki/Theory_of_constraints from Eli Goldratt.

[2] Matt Cholick – https://twitter.com/cholick/status/384816643370532864

[3] Pair Writing: https://pds.blog.parliament.uk/2017/03/29/pair-writing/ and https://gathercontent.com/blog/use-pair-writing-to-collaborate-with-subject-matter-experts

[4] Pair Learning: https://www.infoq.com/news/2018/02/pairing-learning

[5] The original source for Coding Katas is Dave Thomas and Andy Hunts – Pragmatic Programmer.
Our own work on Randori workshops is:

http://agilepainrelief.com/notesfromatooluser/2008/10/tdd-randori-session.html and http://agilepainrelief.com/notesfromatooluser/2008/10/tdd-randori-workshop.html with another example: https://agilepainrelief.com/notesfromatooluser/2012/03/scrummaster-talesthe-team-learn-how-to-learn.html

***

To read this article online with embedded hypertext links, go here:

**https://agilepainrelief.com/notesfromatooluser/2018/06/how-to-cross-skill-and-grow-t-shaped-team-members.html#.XTSAdOhKguU**

# About Mark Levison

**Mark** Levison is a Certified Scrum Trainer and principal Agile Coach of Agile Pain Relief Consulting. With over 30 years of experience in the IT industry, he has been learning and teaching Agile since 2001. Introducing Scrum, Lean, and Agile methods to many organizations, from government departments and banks to healthcare and software companies, Mark has coached professionals throughout Canada.

Mark's training benefits from his study and writing on the neuroscience of learning: *Learning Best Approaches for Your Brain*. He has released one eBook, *Five Steps Towards Creating High-Performance Teams*, authors the blog "*Notes from a Tool User*", and is currently developing another eBook based on his online series, "*Beyond Scrum*."
For more information, visit his website at: www.agilepainrelief.com

# Over-specialization and waste of potential

By Yi Lv and Steven Mak

Posted on February 13, 2018

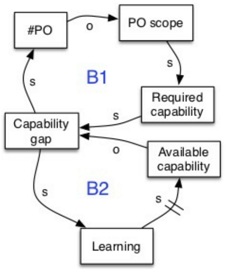
It is natural that people specialize in various things, but when it is too much, it becomes over-specialization. Over-specialization wastes our potential. We shall understand why it happens and how the adoption of LeSS avoids it.

### Eroding goals

"Eroding goals" as a system archetype consists of two balancing loops. Think of any problem as the gap between the goal and the actual. There are two types of solutions. One is to improve the actual so that the gap is reduced (i.e. problem is solved), the other is to lower the goal. Over time, it evolves to keep "eroding goals". It becomes boiling frog. This is very simple but powerful dynamic, and let's see a few examples at work.

### PO specializing in domain

I have described this topic in the article of "team PO as anti-pattern". Here we don't talk about fake POs, who are not responsible for the product. The real PO may still specialize in product domain. This is often a response to the capability gap.

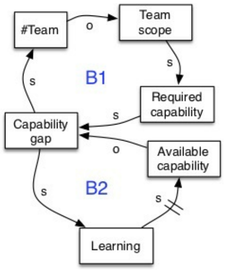


The B1 loop illustrates the solution of lowering the goal. By having more POs, the scope of every PO would be narrower, which requires lower capability.

The B2 loop illustrates the solution of improving the actual. By increasing the learning, the available capability improves. However, it will take time, which makes B1 loop dominant - the goal erodes. This is what we have observed. While growing a company, each PO gets narrower and narrower scope to be responsible for. Also note that we get more and more POs in this dynamic.

### Team specializing in function, component or domain

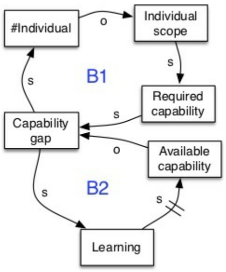
Let's look at team. Team may specialize in function, component or domain. They become functional team, component team and specialized feature team, respectively. Specialized feature team here means that it is feature team and able to deliver feature end-to-end, but it has its own product backlog only containing features in the partial product, i.e. specializing in product domain.



This is essentially the same dynamic as PO specialization. The B1 loop lowers the goal by having more teams and each team responsible for the narrower scope, be it function, component or domain. The B2 loop improves the capability, which takes time. This is why we observe that teams get more and more specialized, and meanwhile we get more and more teams.

### Individual specializing in function, component or domain

Let's look at team members, i.e. individuals. Individual may specialize in function, component or domain too. They become specialists in the team.

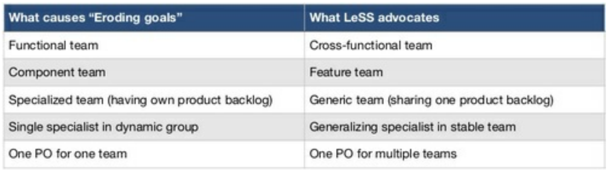


This is actually the more generic case for PO specialization, and we see the same dynamic. Individuals get more and more specialized and they become single specialists. It causes that we have to create dynamic team to match the work, and this dynamic team is feature group or project in matrix organization. Likewise, we get more and more people.

In short, this is something I have observed in many organizations. Over time, people specialize more and more, and the company grows bigger and bigger in size, while people's potential is not fully realized!

### LeSS promotes learning

Interestingly and perhaps accidentally, LeSS avoids "eroding goals".



| What causes "Eroding goals" | What LeSS advocates |
|---|---|
| Functional team | Cross-functional team |
| Component team | Feature team |
| Specialized team (having own product backlog) | Generic team (sharing one product backlog) |
| Single specialist in dynamic group | Generalizing specialist in stable team |
| One PO for one team | One PO for multiple teams |

Which one is better, being comfortable but waste of potential, or "painfully" growing and realization of the full potential? This gets philosophical and surely everybody has its own answer.

***

To read this article online with embedded hypertext links, go here:

https://blog.odd-e.com/yilv/2018/02/over-specialization-and-waste-of-potential.html

# About Yi Lv

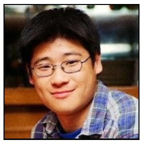
**Yi** Lv (the correct Pinyin is Lü) lives in Hangzhou, China. He is an agile coach at Odd-e. He is the first Certified Scrum Trainer (2008) and the first Certified LeSS Trainer (2018) from China.

From late 2005, while working in Nokia Networks, he started to get acquainted with agile software development, in particular, Scrum. That turned out to be his first experience of LeSS adoption. He led a department inside that product organization and focused on developing teams and Scrum Masters to create sustainability. He left Nokia Networks in late 2010 and joined Odd-e till now.

As a coach at Odd-e, he also worked in other industries such as Internet companies. His focus has been on large-scale product development, especially helping organizations benefit from LeSS and/or its adoption. He has been learning and practicing systems thinking since 2009. Over the past years, he wrote a series of blogs to help see system dynamics in organizational design and change.

He can be reached at yi.lv@odd-e.com.

# About Steven Mak



**Steven** Mak - Coaching and consulting on lean and agile software development.

# IT Leadership Is Broken. Here’s How To Fix It.

By Sam McAfee

Posted on September 28, 2018

IT leadership is broken. The model used by most large and midsize companies is ridiculously outdated, a relic of the 1970s. It’s time for an overhaul.

## Where did the IT department come from?

Long, long ago, computers were the reserved territory of the bean counters. Software usage in corporate America originated inside the Finance department, sequestered mainly to accounting purposes. This is evident in the fact that the first major business application developed for a Graphical User Interface was a spreadsheet.

As applications began to be developed for broader business roles, including word processing and email, it became more common for all or most employees to be issued a computer. Someone needed to keep track of the hardware, networking, and application installation and support for all these machines. And thus, the Information Technology department was born.

The responsibility of IT was thus primarily to provide tech support for business employees, including administering hardware, updating security configurations, and crawling around on the floor under desks with a bundle of Ethernet cables held firmly in one’s teeth.

Then the Internet changed everything. By the mid nineties, led by Bill Gates, Microsoft began aggressively marketing Internet usage, for example bundling Internet Explorer into the Windows OS. In 2000, with the triumphant return of Steve Jobs, Apple quickly followed suit. The result was a massive increase in both personal and business use of online software.

Soon all businesses set up websites, and the more forward thinking leveraged a growing body of Internet enabled applications, what Salesforce later coined SaaS applications. The lowly IT department, typically the only employees who understood technology, suddenly found their duties expanded to include support of the company website and a myriad of new business applications.

As technology gained in importance to all aspects of business, IT departments continued to be seen by most senior executives as a cost of business operations rather than an engine of value creation, despite its new found critical role. As a result, the way that IT projects were planned and funded treated technology as a perhaps unfortunate but necessary nuisance on the balance sheet.

## The Dictatorship of Cap-Ex

The IT project that you are ignoring right now in order to read this article was probably assigned to you from a long list of annually approved IT projects. Each project was submitted for consideration to an IT governance committee. The approval of the project required a compelling business case to be submitted, containing all of the expected business value of the project for the next five years, an exacting list of expected costs down to the hourly rate of individual contractors, and a precise timeline for execution, release, and maintenance.

The regrettable reason for this type of project approval process is the tyranny of the finance department, and in particular corporate accounting practices. From the left hand side of the CEO, Finance still rules over IT with an iron fist, ensuring that each and every IT project justifies itself by keeping its footprint on the quarterly earnings report as minimal as possible.

With this mindset, Finance departments have been legendary in tightly constraining IT budgets, primarily using the weapons of capital vs operating expense classification of costs. This little bit of accounting arcana exerts an extraordinarily dampening influence on the ability of traditional businesses to take advantage of the increasing acceleration of digital transformation. It’s likely one of the biggest factors keeping your business from being as flexible and adaptive as your growing list of competitors.

First, the classification of software as a fixed capital asset is a total farce, a quaint anachronism from the days when software was installed using a phased-gate “waterfall” process, and expected to continue running unchanged more or less indefinitely. The only way that software makes sense as a capital asset is if it can be sold to another firm the way that manufacturing equipment and other capital assets can be accounted for in the price of an acquisition.

Software isn’t like that anymore. There is no application in your company running right now that is worth anything to anyone outside your firm. Software applications are platforms for facilitating the delivery of value to your customers. They are simply the automated manifestation of your business model.

Let me put this a bit more strongly. Insofar as CapEx reporting of investment into software development makes your balance sheet look better, it is essentially a lie in accounting terms. Your tax picture may look rosy to Wall Street analysts. But it comes at the cost of a clear realization of the role of software development as a core operating cost in your overall balance sheet. In short, you're fooling yourself.

Further, the sheer administrative weight on IT managers to classify labor costs on all IT projects as either capital or operating expense severely hinders and dis-incentivizes IT teams from running software development projects in either an Agile (read: fast and flexible) method, or leveraging learning and continuous improvement by releasing “minimal viable products” (MVPs) as a core competency of IT.

I am certainly not the first to advocate dropping this obsession with CapEx. Lean expert Mary Poppendieck wrote an excellent piece on the subject last year (see: http://www.leanessays.com/2017/11/the-cost-center-trap.html). “Lean Enterprise” authors Barry O’Reilly, Jez Humble, and Joanne Molesky have also helped popularized this idea.

## Antiquated Project Management Approaches

The impact of Cap-Ex thinking in IT extends into every nook and cranny. As I’ve written extensively in earlier articles, the common approach to IT project management is to assemble a “team” by cobbling together resources from a vast matrix of skills and availability.

Projects in all too many firms continue to be planned and executed in an old-fashioned “waterfall” approach. With fixed budgets, scope, timelines, and resources, IT projects are extremely sensitive to unexpected shocks and likely to quickly run over time or over budget with the slightest provocation. The adoption of Agile methodologies, so desperately needed in most firms, directly contradicts the Cap-Ex mindset, and thus almost immediately runs into problems.

To make matters worse, companies continue to limp along with outdated and antiquated infrastructure. While nimble tech giants like Netflix and Amazon are able to deploy literally thousands of small changes to their software every day, the majority of traditional firms struggle to get a working and bug-free release out the door once a month. This staggering difference in speed of iteration is a large part of the reason Amazon continues to conquer one market vertical after another, a seemingly unstoppable juggernaut. Without investing heavily and quickly in flexible cloud-based infrastructure, it is highly unlikely your firm will survive once Amazon sets its sights on your market.

## Fighting Fire With Fire

Look, Amazon is eating every industry in the economy, one by one. Should you find yourself on the defensive when they are already entering your market, it is already too late. Even if your market is low on their priority list, there are a hundred tiny startups coming after you anyway.

The time to prepare for the inevitable wave of disruption is now. Since what you’re doing to innovate is totally not working, perhaps you’d be willing to try something different.

## Accounting for Innovation

For starters, abandon the notion that any digital asset in your company is a capital asset. Investment of software lands squarely on the operating costs side of the balance sheet. You’d be much better off spending your time finding ways to empower your software teams to make up for that operating cost by producing more value for customers more quickly, rather than building assets that depreciate. Accounting tricks are not going to hide the truth about how inefficient you are at competing digitally--so stop trying.

## Teams, Products and Services

Stop thinking of your IT personnel individually, and start thinking of “teams” as your primary unit of people. Solid, cohesive, collaboration is the key to increasing your throughput of value to customers. And that collaboration comes from having well-exercised teams.

Further, stop planning IT “projects.” You have only two IT work units in your firm, products and services. Products are essentially just the digital front-ends where your customers or internal employee users access the business processes enabled by your company. Services are the backend business capabilities that power those products. All teams should be organized around supporting either a product or a service.

## Build Product Roadmaps From Strategy

What your teams are supporting should align with the company strategy. Yes, you may be doing something now that sounds like this. However, it needs to be flexible and adaptive, not a fixed annual list of “IT projects”. IT leaders should be able to derive and articulate concrete IT objectives from the set of overall business objectives and corporate strategy formulated by executive team.

Then, your feature roadmap for each product and service that you support should roll up into this overall list of IT objectives. Teams that support those products and services can be aligned to, and incentivized by, their ability to hit those objectives. No more biasing the production of outputs by individuals. Now they’re measured against their ability to execute the strategy of the business as a team.

## Invest in Infrastructure and Automation

If your deployment pipeline is replete with locks and gates due to overly conservative system administrators operating a dinosaur operations infrastructure, it is time to consider heavily investing in the cloud.

The startup down the street is building their software in such a way that each new engineer who joins the team can be assigned a laptop, clone the codebase from Git, and get the application running in minutes with a few commands.

The suite of automated tests ensures that each code push can be exercised automatically and rejected if necessary. This gives software engineers the confidence to move with all deliberate speed from a feature in the backlog to a working and welcome addition to the production system used by customers. To compete with teams like this, invest now in infrastructure and test automation.

## Build an Effective Recruitment Pipeline

It’s hard to find great technical talent. That’s the key complaint I hear from IT leaders these days. There are two main obstacles in being able to attract fresh new software engineering talent to older traditional companies. The first is the culture. The second is the tools.

The culture in IT in traditional companies is, as mentioned above, overshadowed by the tyranny of Finance and the damage inflicted by CapEx thinking. As such, no self-respecting graduate from Stanford’s computer science department is going to choose to go to work at your company over a place like Google or Facebook, without being darn sure that your company embraces a culture of rapid experimentation and innovation.

Even if your culture is amazing, if you’re software tools are ten years old, you are not going to attract the best and brightest. Can a new hire at your company use the same technology AirBnB uses to build their web interfaces for their next project? Are the service engineers able to build data pipelines and self-healing infrastructure like Netflix uses for its systems? If you answered no, you'd better start thinking about how to change that.

Building a recruitment pipeline is a reciprocal solution to this problem. In order to attract talent, you need to make some hard but necessary changes. In order to implement those changes, you'll need to recruit new talent. The two go hand in hand. So start thinking about how to make just enough of the culture and tool changes to attract new talent, and find ways to get out in front of audiences and talk about it.

## Invest in Training and Professional Development

Maybe your IT personnel lack the chops to keep up with the startups massing at the gates. But that's not because they don't want to learn new technologies and approaches. People get into technology in the first place because it's a way to constantly learn new and exciting things. You'll be more likely to keep your valuable employees if you invest in their long term development.

My company, Startup Patterns, offers a range of great workshops that help transform old-fashioned IT teams into nimble engines of innovation. And we're certainly not the only ones out there. Seek help from outside sources. All too many companies I encounter have tunnel vision, and fail to get fresh ideas and insights from competitors in their fields. Send your people to conferences. Heck, encourage them to speak at conferences!

I've been working in IT for two decades now. It's full of great, reliable, smart, hardworking folks. We just need to unleash that creative potential so that they can leverage their skills to really transform their businesses.

—

## How are your teams doing?

Take our free product organization self-assessment and find out! Take the assessment here.

***

To read this article online with embedded hypertext links, go here:

**https://www.linkedin.com/pulse/leadership-broken-heres-how-fix-sam-mcafee/**

# About Sam McAfee

As a Silicon Valley veteran of 20 years, Sam has helped many companies build and scale new digital products, find product-market fit, and build and improve product development capabilities. He works with product executives and teams to build strong leadership and execution skills, through a focus on mindfulness, autonomy, and social impact. He has worked with companies large and small, including Adobe, Sharethrough, Teach for America, Anthem, PG&E, Blackstone, Credit Sesame and many others. He authored the book, *Startup Patterns*, released in 2016, and is a regular speaker at product and engineering events around the US.

# Accelerating Scrum Success with Lean Principles

By Nirmaljeet Malhotra

Posted on October 4, 2018

THE END JUSTIFIES
THE MEANS

For organizations that have been on the agile journey, Scrum has been the framework of choice for some time now. The simplicity of the framework and its ability to enable the empirical process are some reasons behind scrum's success. Having said that, one may question if teams have truly attained agility by doing Scrum.

The terms scrum and agile are often used interchangeably which is where the distinction between doing Scrum and becoming agile becomes important. According to me, **if Scrum is the means, agility is the end**. To put it simply, Scrum is like a driver's manual. A learning driver uses it till the time he/she is internalizing the rules of driving on the road. Once done consistently over a period of time, the driver is easily able to navigate through the traffic without referencing or following the manual at each step.

## From Doing Scrum to Being Agile – Impediments

Every aspect of scrum has a purpose. From the roles, artifacts to the events, there is a value associated with everything. However; it is noticeable that teams that have been doing Scrum for a long time have not always shown behaviors or patterns of a high performing team. This points to the issue of scrum being adopted as a process without the end in mind. It has been validated the time and again during my engagements with organizations that have been doing scrum for years.

Agile is consistently referred to as a mindset based on the values outlined in the agile manifesto combined with the agile principles. However; teams struggle to successfully attain the mindset shift for the reasons mentioned below:

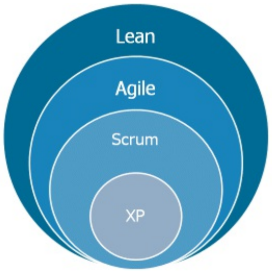
- **Training(s)** – Most agile adoption initiatives start with trainings which are intended to familiarize teams with Scrum framework with limited focus on the agile mindset. Yes, the agile manifesto and principles are covered but there is not enough focus on educating the teams on how they can graduate from doing scrum to attain agility. This often leads the team(s) to believe that Scrum is the end state.

- **Scrum as a process** – Once a team has been trained, the next step is to ensure adherence to the framework. Generally team will have a Scrum Master whose job is to promote and support Scrum as defined in the scrum guide. Interestingly, while some organizations look at their Scrum Masters as coaches, **the scrum guide does not associate the term "continuous improvement" with the Scrum Master role**. This also makes the teams assume scrum as a process and end state.

- **Process compliance** – Organizations with multiple teams often enforce compliance which is attributed to the need for consistency  of process and the usage of supporting tools like Jira or Rally. This idea of compliance is generally directly in conflict with the attributes of self organizing and self managing teams forcing the teams to live in the scrum box.

Above mentioned are just some of the reasons. You might have your own but the focus is to emphasize the need to go beyond the process.

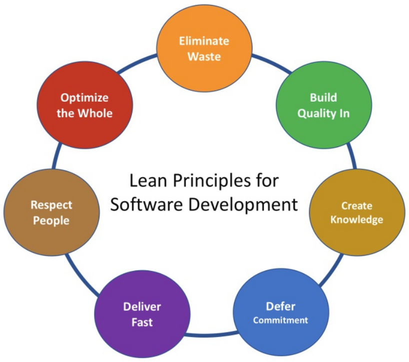
## Applying Lean Principles



There is knowledge base that connects the roots of Agile to Lean. Lean got its start from manufacturing and applying it to knowledge work required a mindset shift. Lean introduces a customer oriented flexible system for software development. Applying lean principles with scrum teams have shown some encouraging results.

In their book, **Lean Software Development: An Agile Toolkit**, Mary and Tom Poppendieck outlined how these Lean principles can be applied to software development. In the most simplistic form, these lean principles can be applied on top of scrum the accelerate the move from doing scrum to being agile.

Lean and agile be definition are different things but they are great partners intended to achieve common outcomes. Here is how lean principles can be applied to further mature scrum teams:



**Eliminate Waste** – A variety of wastes impact the outcomes of scrum teams. In large organizations, these wastes are commonly attributed to processes, complexity, structures, silos, hierarchies etc.. Organizations and teams that are relentless about continuous improvement can benefit greatly from consciously and incrementally looking for opportunities to eliminate waste. Value stream mapping should be conducted at the business and development process level to expose waste and create urgency to eliminate them.

**Build Quality In** – One of the common struggles for scrum teams is to deliver the "potentially shippable" artifact. Reasons include quality of requirements, structure of requirements (not vertically sliced), dependencies, unstable environment, team silos, processes, handoffs, metrics and more. However; the most common reason is the number of defects produced in sprint or the lack of tools and/or automation to support end to end testing and the mindset of dealing with defects.

Taiichi Ohno when talking about the Toyota Production System talks about jidoka (self-regulation). The idea came from a loom which would stop automatically when a thread broke. When applied to software development, the idea is to enforce behaviors of urgency, collaboration, swarming etc.. as soon a defect is found.

Additionally, the lean principle of building quality in starts by suggesting that quality if everyone's job and not just QA's and it needs to be a disciplined practice. Lack of it also is in conflict with the first principles of eliminating waste.

**Create Knowledge** – This is referred to as "Amplifying Learning" in Mary and Tom's book where the best way of learning something is by doing it or in other words by actually creating value. Lean is also about learning through experiments. These behaviors and/or mindsets are not often observed with scrum teams.

Some common anti patterns that impede the principle of creating knowledge with scrum teams include team structures, where teams are build up of many specialized skills thus creating silo success criteria and outcomes. Additionally, scrum teams can also been seen doing "foundational" work with an excuse to build a sense of predictability. This is precisely why agile relates the term "evolving" to all dynamic aspects of software development including requirements, architecture, design etc..

**Defer Commitment** – While scrum revolves around the idea of short feedback loops, teams are often challenged with timelines or scope or sometimes both for a variety of reasons. This is commonly observed with organizations that limit agility to the teams but still have maybe a PMO that engages in yearly planning activities thus leading to assumed commitments.

Similarly, at the team level, the inability of the team to meet it's sprint commitment (even though the term "commitment" has been taken out of the scrum guide), can go against the team leading to compromise in other areas, more often quality.

The lean principle of defer commitments does not mean that the teams do not plan or make uninformed decisions. It encourages that decisions be made at the **last responsible moment**. Last responsible moment can vary from companies to industries to teams but the basic idea revolves around acknowledging and conducting experiments to enable informed decision making.

**"In preparing for battles, I have always found that plans are useless, but planning is indispensable" – Dwight Eisenhower**

**Deliver Fast** – **"Ability to deliver fast"** is the most common reason why most companies adopt agile methods. Fast can be interpreted differently in different context. One example is **Etsy** that delivers code into production multiple times each hour. But not all businesses are the same.

The term "fast" was originally intended to enable fast feedback loops to allow teams to inspect and adapt in their pursuit to cause customer delight.

Every team wants to deliver fast and get value in the hands of the customers as quickly as possible, but most scrum teams are unable to do so for a variety of reasons. Some examples:

- Organizational and team structures leading to increased complexity
- Lack of supporting practices and tools
- Big increments; looking too much into the future
- Lacking urgency to remove impediments
- Intent of building a perfect solution

Going back to the principles mentioned earlier, deliver fast can be the outcome of eliminating waste, building quality-in and creating knowledge. A focused effort to apply these principles leads to the lagging outcome of faster delivery

One of the concepts you'll hear referred to a lot in Lean is **"concept to cash"**. It refers to the lead time from the time the idea was conceived to when the customers start purchasing it, or it can start adding value by saving us costs, reducing costs, etc.. as fast as possible.

**Respect People** – The 5 values of scrum (**commitment, courage, focus, openness and respect**) help build trust within the team and the scrum team members are expected to learn and explore these values as they work with the scrum roles, events and artifacts. However, a common observation is about how most important decisions are taken outside the team that have a direct impact on the team. Simply put, decisions are taken away from the where the work happens.

Another challenge for teams is the issue of psychological safety. In the lean world, respect is all about developing and empowering the people and trusting them to do the right thing. Lean talks about this idea of a "Gemba", which refers to the process of leaders going and observing where then works happens, conducting enquiries and identifying counter measures for problem solving along with the team.

Lean also encourages respect for people by communicating proactively and effectively, encouraging healthy conflict, surfacing any work-related issues as a team, and empowering each other to do their best work.

**Optimize the Whole** – Sub-optimization is a serious issue in software development. Mary and Tom point to 2 critical reasons behind the sub-optimization. First is where developers release sloppy code for the sake of speed and second is the long cycle time that is created between developers and testers as a result of handoffs.

Leans suggests use of value stream mapping to design, produce, and deliver a product or service to customer. After identifying how value flows through their teams, many organizations decide to organize their software development teams to be complete, multi-disciplined, co-located product teams, which enables them to have everything they need to deliver a request from start to finish, without reference to other teams.

***

Scrum is a framework which means it is an essential supporting structure or the basic structure underlying a system. Teams that are comfortable with scrum and achieve the expected outcomes can apply the mentioned principles are various levels to quickly transform themselves into a high performing team.

Regardless of which framework the team chooses to adopt, it’s important to understand the principles behind the method in order to ensure a sustainable, disciplined practice. If your team is doing Scrum but is not consciously implementing agile and/or lean principles, the outcomes can be slow and the journey long.

***

To read this article online with embedded hypertext links, go here:

**https://nirmaljeet.com/2018/10/04/accelerating-scrum-success-with-lean-principles/**

# About Nirmaljeet Malhotra

**Nirmaljeet** Malhotra is a passionate agile, product, lean and leadership coach. He has been part of multiple small and large agile adoption and transformation journeys and has focused on continually improving his coaching skills to enable organizations succeed with change.

In his present role, Nirmal works for Improving in Dallas and a consultant enterprise coach with Intuit.

Nirmal holds a Master’s degree in computer science and many certifications in agile, coaching and leadership. He likes to share his experiences and thoughts through his blog nirmaljeet.com. He is also a frequent speaker at conferences and meetups.

Nirmal can be reached at nirmaljeet.malhotra@gmail.com

# Dear Beloved Clients, Please Start By Doing

By Kathryn Maloney

Posted on July 16, 2018

The work starts only when your hands get dirty.

I've recently had conversations about the inherent values and differences between our approach at The Ready and an Agile approach to transforming the way a system and teams operate. I am admittedly impatient around comparisons like these for a few reasons. The first one is that what pretty much always sits behind them is the reptilian brain wanting to negotiate risk, fear, and impotence veiled as an intellectual exercise around practice and methodology. I get it and am empathetic to it, and I also have to name it. A second reason is that I am a fan of good and ethical practice, *however* it is packaged or named. We aren't the end all, be all nor is the other. But, our methodology and practice are very, very good. And, the other may be as well?

And lastly, I'm a person and practitioner who has a far deeper commitment to leading the experience of change over debating and cognating about change. The reason for this is that if you are wondering about changing, most likely you are in need of changing. To get you changing, we have to begin activating it — which *doesn't happen* by talking about it.

> Too much time and money is spent considering change versus just doing it — and I am far happier to move you.

So, here are some thoughts in response to the question of our OS canvas versus Agile, but more importantly to the questions behind the question.

We know that every organization has an operating system. It's made up of the countless assumptions, principles, practices, and behaviors that manifest into culture. Adaptive and resilient organizations understand that being deliberate about their way of working is an essential and continuous process of leveraging tools, practices, and rhythms to root (1) thinking strategically, (2) working dynamically and iteratively, and (3) learning constantly. We developed the Operating System (OS) Canvas as an underlying framework to provoke systemic thinking, conversations, and reflections about all elements of an organizational system. The canvas provides scaffolds to consider and act on cultural, behavioral, and structural shifts — and illustrates how they knit together to create a living, breathing, constantly evolving whole.

Some parallels within our OS canvas to Agile practices are obvious — and integrated. Agile methods (along with other methods like Lean, Kanban, Open Space, and the canvas) can all be considered parts of a broader ways of working container of org design practices. When we teach teams new meeting and teaming structures for instance, people familiar with Agile will certainly feel the commonalities because we draw from many methods in our designs, including Agile practices. And, I'm guessing any Agile transformation effort is likewise pulling from the deep well of systems change resources. Where we are cautious (and even skeptical) — with our own canvas methodology and any other method or tool such as Agile — is to keep people from falling into the ideology trap. Commoditizing any method or practice as a whole system change *ideology* (versus a *method of intervention*) will quickly create limitations on their application in complex, perpetually changing ecosystems. All are ways in to enliven a system to be and do differently. None should be sold, promised, or practiced as end states.

> Designing a new OS means using the underlying framework to guide new understanding, thinking, learning, and implementation while making it distinctly your own and deliberately getting better as people, a team, and an entire organization every day.

It isn't a fixed system to learn and apply, because it is not an *ideology*. Rather, our OS canvas is a method to think holistically about a system's principles and practices in its current state combined with a navigational map to adopt new tools and practices considering the interface of *work that needs to get done* and *how to get it done* in the most thoughtful, relevant, fast, and coordinated way. In designing a new OS, you must experience and learn new tools and techniques and then learn to apply them bravely (because it takes bravery to be/do differently), dynamically (because we live in a world of work and leading that demands of-the-moment choreography), and consistently (because designing a new OS is foundational and perpetual to compete and thrive in today's fast-changing, globally connected world.) To do this, we prefer to get into the work quickly — rather than talking conceptually — to create context and *be in the experience* of learning by doing. It may be disorienting initially, yet scales change *much* quicker.

Org design, agile, lean, holacracy — or any so-called transformation work are all mindset shifts at the end of the day. They are not destinations and frankly, rarely is there an arrival. Readying to scrutinize your personal, team, and organizational mindsets is an enormous part of the work and deeply important to adoption of a new context from which you work, learn, create, and thrive.

> We can teach people new meeting structures, decision tools, communication technology, teaming structures, etc. all day long, but if you aren't prepared to give up what you know to make room for what you don't yet know — and feel slightly off balance in the process — much less new and different will occur.

The reason being that it's simply impossible to change without changing. Talking about changing is merely delaying effort. We can and will provide the tilled soil, but you have to sow the seeds. And that understanding and motivation has to start right from the beginning of considering a change effort.

Here are a few goalposts for embarking on the mindset shifts an OS design and new ways of working journey will ask of you.

## Experiencing is believing

Talking about change is like writing about beautiful fashion without the visual or describing an amazing meal or beautiful wine without tasting it. Seeing and feeling are far more powerful and emotional — and doing / scaling change lives in the actual experience. Show rather than tell. Put down the megaphone, stop wordsmithing, debating or opining, and just do. People will feel the commitment and glow, and it will naturally spread.

## Don't wait for permission

While org change and transformation rhetoric became steeped in top-down sponsorship *and* while having strong leadership ringfence change efforts is certainly super helpful, it's not an imperative. *Small groups of thoughtful citizens can change the world.* They've been doing it forever. Declare your independence, step into your personal authority, and show people the way. Make people curious and take notice rather than wait for permission. *This is leading*.

## Prepare to lose in order to gain

Making space to learn new ideas and develop new muscles is an imperative. Otherwise, it’s like dieting while eating all the same bad things or moving without dropping stuff off at Goodwill. Shedding is a natural part of evolution. Gripping and attachment exist to create friction around growth and change. Resistance is what causes adaptation energy. Lean in and *let go*.

## Mind your ego

The monkey mind will trick you at every turn into believing that *it is unsafe to try new things; stupid to not drive, drive, drive rather than create space for bigger thinking, deeper connecting, and reflective learning; dangerous to not know all the possible pitfalls before trying; and downright ignorant to show vulnerability, fault or (god-willing) take a risk*. Leading means learning to not be dragged down the street in the shackles of your ego and instead settle into the very humanity that is you. Take off the veil, quiet the noise, and let *you* radiate.

## Stop planning and start doing

Project plans have been an outdated methodology for at least a decade, if not longer. We live in a time where anticipating, adapting, and pivoting are three of the most critical leadership skills needed in any role, in any organization, in any industry. Test and learn your way into next steps rather than falsely believing you can predict and plan your way to innovation. Set a direction, but steer continuously. Everything around you will change far faster than your project plan allows for so use tools that enable and support responsiveness. Think *wayfinding over navigating*.

## Be grateful and present

Learning from others and about ourselves, being on a team with good humans, and having the opportunity to contribute to something bigger are nothing short of gifts. Don’t artificially or passively be grateful. Look for moments that kick your ass or even peel back the slightest layer of new awareness or thinking — and feel the gratitude for being alive and present to the experience. Say thank you. Tell people you love them. Live the moments consciously. It’s *contagious*.

None of the above is easy, fast, or linear which maps perfectly to understanding the complexity of systems. Whether it is work framed by the OS Canvas or Agile — or anything else — nothing is a panacea. You will need to step in and *do work*. The choice has to be about braving learning, experiencing yourself differently, and being in the truth of how systems, change, and work actually work — and using whichever tools, practices and methods that enable that. Sometimes, you may use more than one method over time or even two methods at the same time. Fantastic. If it serves to create and leverage change as energy and toward a better understanding of how to lead, operate, and organize from a perpetual state of change you are doing *work that matters*. Just start by starting making sure you are doing real work. You’ll know because it feels challenging, personal, enlivening — and yet not ideological. Then, just keep evolving from there.

---

Did you know that The Ready founder, Aaron Dignan, recently finished a new book all about how organizations can change the way they operate? If you’re interested in the future of work movement, the work we do at The Ready, or simply how to make your own organization better, then you should definitely order your copy of *Brave New Work* today.

***

To read this article online with embedded hypertext links, go here:

**https://medium.com/the-ready/dear-beloved-clients-please-start-by-doing-not-thinking-8dff9dd0ac98**

# About Kathryn Maloney

**Kathryn** is an advisor, coach, and consultant working with organizations large and small to design systems, teams, and approaches that enable fluency, fluidity, and function amidst complexity, technology, and the need for more humanity. She spends a lot of her time working with leaders to bridge organizational worlds, transitioning from former leadership paradigms and becoming more comfortable, adaptable, and resilient in current ones. Understanding and applying methodologies and technologies without holding too tightly to them and projecting unreasonable expectations onto them is a large part of her professional charm. Having worked for over twenty years in the spaces of systems change, strategy, and org design with companies as large as GE, BCG, and Boeing, government agencies like the Federal Trade Commission, Federal Reserve, and HHS - and as small as start-ups in their infancy, Kathryn is cautious about this moment in time and the rhetoric around agile - or any other promise - as a one-size-fits-all solution. She has an academic background in the applied behavioral sciences, is conditioned with a systems thinking lens, influenced by the org systems pioneers of the 20th century, and trained as a qualitative researcher, leadership coach, and process consultant for organizational (as opposed to family) systems. She is a founding partner of the org design consultancy The Ready and runs her own practice www.kathrynmaloney.com. Kathryn lives in New York City. Follow her on LinkedIn to capture interesting writing, the pollination of ideas, and cool events.

# The Tragedy of Craftsmanship.

By Robert C. Martin (Uncle Bob)

Posted on August 28, 2018

*You look pensive.*

Yes. I just read a transcript of Martin Fowler's keynote at Agile Australia, 2018. He called it The State of Agile in 2018.

*Ah, yes, a great talk.*

- *Beware the Agile-Industrial-Complex.*
- *Maintain Technical Excellence.*
- *Products over Projects.*
  *Great stuff! So what bothers you?*

In that talk he said that the formation of the Software Craftsmanship movement was tragic.

*Yes. He's right.*

He is? How could that be? I thought Software Craftsmanship was a good thing.

*Oh, it is. It's a very good thing.*

But then why…?

*The tragedy is that the Agile movement was supposed to promote the ideals of Craftsmanship; and it failed. Horribly.*

I don't understand.

*The Agile movement got so involved with promoting conferences and with certifying Scrum Masters and Project Managers that they abandoned the programmers, and the values and disciplines of Craftsmanship.*

But I thought it was programmers who started the Agile movement.

*Yes. It was. That's the grand irony. It was programmers who started the Agile movement as a way to say: "Hey look! Teams matter. Code should be clean. We want to collaborate with the customer. And we want to deliver early and often."*

*The Agile movement was started by programmers, and software professionals, who held the ideals of Craftsmanship dear. But then the project managers rushed in and said: "Wow! Agile is a cool new variation on how to manage projects."*

*There's an old song, by Alan Sherman, called J. C. Cohen. It's about a subway conductor who did such a great job at pushing people into the train cars, that he pushed the engineer out. This is what happened to the Agile movement. They pushed so many project managers in, they pushed the programmers out.*

That's not quite the way Martin Fowler described it. He said that the Craftsmanship movement started because a bunch of programmers said: "Oh, we need to create a whole new world for ourselves […] where we can go away, get away from all of these business experts and project managers and business analysts, and just talk about our technical stuff."

> *Oh, no. Martin got that completely wrong. It's very clear from the Software Craftsmanship Manifesto that the goal of Craftsmanship is to continue and expand the Agile message. Software Craftsmanship is not some kind of Techie Nocturnal Emission. Software Craftsmanship is simply a continuation of the original goals of Agile.*
>
> *Craftsmanship is the Agile, that the Agile movement left behind.*

Left behind? Left behind to do what?

> *To promote conferences, certifications, and fancy new project management strategies.*

What's wrong with certifications?

> *Let me put it this way: Anyone who suggested a two-day course to certify Craftsmanship would be laughed out of the room, laughed out of the town, and laughed out of the state. The very idea is absurd.*

OK, but how can you have a movement without hype, certifications, the training, the conferences? Don't you need that stuff to get people's attention?

> *Perhaps. But I hope the Craftsmanship movement doesn't leave it's original purpose behind the way the Agile Movement did.*

What purpose is that?

> *The original Agile purpose. You see, Craftsmanship is not about new stuff. Craftsmanship is about old stuff. It's about working well, adding value, and doing a good job. It's about interacting, communicating, and collaborating. It's about productively adapting and responding to change. It's about professionalism and ethics. It's about the goal that Kent Beck had for Agile.*

What goal was that?

> *At the Snowbird conference, in 2001, where the Agile Manifesto was written, Kent Beck said that one of our goals was to heal the divide between programmers and management.*
>
> *The Agile movement abandoned that goal by turning Agile into a business that promotes "new-and-better" ways to manage. Instead of bringing managers and programmers closer together, the Agile movement focussed almost entirely on project management, and virtually excluded the programmers.*

And so that's why the programmers split away?

> *No! The programmers did not split away. The programmers stayed the course! The programmers continued to pursue Agile as it was originally conceived. Read the opening line of the Agile Manifesto: "We are uncovering better ways of developing software by doing it and helping others do it." It is Software Crafts-men and -women who are continuing that work. It's not the project managers in the Agile movement. They're off pursuing something else?*

What is it that they are pursuing?

> *Newness and Novelty. Nowadays the Agile movement is about "The Next Big Thing" and "The Bold New Idea". They need novelty to keep the enthusiasm and energy high. They need that so that people sign up for conferences and certifications. They need to be seen as making – "progress". Agile has become a business; and businesses need to grow.*

It looks to me like they are succeeding.

> *They are. They just aren't succeeding at the original goals of Agile. They split away from those goals in order to feed the need for Novelty and Newness. The result, unfortunately, has been what Fowler and Jeffries have called: "Faux Agile", "Dark Scrum" and "Flaccid SCRUM".*

This is all a little hard for me to believe.

*Let me prove it to you. What was Fowler's first point in that talk of his – the point about the Agile-Industrial Complex?*

He said something to the effect that people work best when they choose how they want to work.

*Right! On a software development team, who does most of the work?*

Well, programmers of course.

*How many programmers were attending Fowler's talk?*

Well, he said it was "a smattering", "very few", "very much a minority".

*QED. Who goes to Agile Conferences? Not programmers. Not the people who do the bulk of the work. Programmers started those conferences. Programmers started the movement. Programmers don't go anymore. It's not the programmers who have changed. It is the conferences, and therefore the movement, that has changed. The Agile movement moved away from the programmers – from Agile. QED*

But…

*Look. Agile was never about project management; but that's what they've turned it into. Agile and project management are utterly orthogonal things. Agile is not a better way to manage a project. Agile has nothing to do with managing a project. Agile is a set of values and disciplines that can help a relatively small team of software crafts-men and -women build small to medium sized products.*

But isn't that management?

*No! God No! Project Management is about dates, and budgets, and deadlines, and milestones. It's about personnel management and motivation. Good management is absolutely necessary; but it has nothing whatever to do with Agile.*

*Here. Look at the Agile Manifesto. Notice those four statements, and how they are divided between left and right. What divides the things on the left and right from each other? The stuff on the right is management. The stuff on the left is Agile. Managers invoke processes and tools. Individuals on Agile teams interact. Managers drive comprehensive documentation. Agile teams build working software. Managers negotiate and manage contracts. Agile teams collaborate with customers. Managers make sure plans are followed. Agile teams respond to change.*

But aren't Scrum Masters kind of like project managers?

*Heavens no! Scrum Masters are coaches, not managers. Their role is to defend the values and disciplines. Their role is to remind the team of how they promised themselves they would work. The role was supposed to be shared by the team, not usurped by managers. Every few weeks a new team member would volunteer to act as coach – if needed. The role was supposed to be temporary. A mature team doesn't need a permanent coach.*

Wow, that's sure not what they teach now. So I guess you think Agile is just ruined then.

*No! Agile is alive and well, and thriving in the Craftsmanship mindset. That's where Agile relocated when the project managers invaded and took the Agile movement over.*

So then, what is the Agile movement?

*Nowadays, the Agile movement might as well be an unofficial branch of the PMI. It's a business that promotes conferences, training, and certifications for project managers. As such, it has become antithetical to Beck's original goal. The Agile movement does not heal the divide between programmers and managers; it exacerbates it.*

It seems like you are saying that the Agile movement isn't Agile.

> *It's not. It gave that up long ago. Nowadays the Agile movement is about the horribly flawed idea that project management is what makes a team Agile.*

Well, isn't it?

> *No, no, not at all. You see, an Agile team is a group of crafts-men and -women who hold the values and disciplines of Agile dear. An Agile team will be Agile no matter how the project is managed. On the other hand, a team that is not Agile will not become Agile simply by virtue of a new and fancy project management strategy. Such a team will be Faux Agile.*

Are you saying that a good manager can't lead a team to be Agile?

> *It is a rare manager who can inculcate the values and disciplines of Craftsmanship. It's not impossible; but it's not common. Agile teams are most often composed of people who already share the values and disciplines of Agile – of Craftsmanship. Thinking that a team can become Agile simply because a Certified Scrum Master is the project manager is a pipe dream.*

So then what's the future?

> *The future is what it has always been. The values and disciplines of Agile will continue to help relatively small software teams build small to medium sized products, and will help to heal the divide between programmers and management. Today, those values and disciplines are held by people who, whether they know it or not, align with the ideals of Software Craftsmanship.*
>
> *I don't think we need an organization to promote Craftsmanship. I don't think we need a "Craftsmanship Alliance". I think all we need are people of good will – individuals who interact and collaborate – communities of professionals who work to promote change by steadily adding value. I think the ideas of Agile – the ideas of Craftsmanship – are robust enough to grow and spread without an organization to drive them.*

So then Software Craftsmanship was not a tragedy?

> *How could the ideals of Craftsmanship ever be considered tragic? They are eternal ideals that humans have aspired to for as long as their have been humans. The tragedy was that the Agile movement became a business that left the original values and disciplines of Agile behind.*

***

To read this article online with embedded hypertext links, go here:

**https://blog.cleancoder.com/uncle-bob/2018/08/28/CraftsmanshipMovement.html**

# About Robert C. Martin

**Robert** C. Martin, aka, **Uncle Bob** has been a software professional since 1970 and an international software consultant since 1990. In the last 40 years, he has worked in various capacities on literally hundreds of software projects. In 2001, he initiated the meeting of the group that created Agile Software Development from Extreme Programming techniques and served as the first chairman of the Agile Alliance. He is also a leading member of the Worldwide Software Craftsmanship Movement - Clean Code.

He has authored "landmark" books on Agile Programming, Extreme Programming, UML, Object-Oriented Programming, C++ Programming and most recently Clean Code and Clean Coder. He has published dozens of articles in various trade journals. He has written, directed and produced numerous "Code Casts" videos for software professionals. Bob is a regular speaker at international conferences and trade shows. Mr. Martin is the founder, CEO, and president of Uncle Bob Consulting, LLC and Object Mentor Incorporated.

Uncle Bob has published dozens of articles in various trade journals, and is a regular speaker at international conferences and trade shows.

# Use This 1 Method to Do More Work in Less Time

**Discover how an Agile approach can help you get to market faster.**

By Maria Matarelli

Posted on August 21, 2018

Agile approaches to completing work have been known to help people to do more than twice the work in half the time. What could that kind of productivity increase do for your business?

Let's compare what is different in using an agile approach from traditional project management. A traditional project management approach focuses on gathering all requirements up front, making sure everything is planned out and designed thoroughly and then using a phased approach to develop, test, and launch a product to market.

Think about launching a website for example. The challenge with this traditional approach is that it may take a long time to get all of the information together for every single page on the site before going live. However, you don't really need every page fully completed in order to launch. If you wait several months for a website to be developed, you are missing out on having that online presence and could be missing a lot of business by people not being able to find you and your services.

An Agile approach to launching a website would be to identify a smaller chunk of work that could be completed as a Minimum Viable Product (MVP). This could be the home page with a nice photo, some information on your business hours, an overview of what you do and maybe a contact phone number. You could probably launch that home page within a week. Then you can build upon that and add a full services page the following week. Then add another page the week after.

The benefit to breaking the work down into smaller pieces is that you can release something to market that is fully developed and tested while ensuring that you have quality built in.

You don't need to know everything to get started. In fact, often times, things may change based on the feedback that you get from your first release. This can even lead to a better end product by taking the time to incorporate real customer feedback.

## Break Things Down Smaller

When you try to launch a large project, it can be challenging. You may run into roadblocks or delays that you hadn't anticipated. This applies to any initiative that feels big and overwhelming. You may even feel stuck at the beginning and not know where to begin.

When you take a larger project and break it down into smaller pieces, such as just looking at what it would take to complete the home page for a website, suddenly, it can seem more manageable. You may not feel as overwhelmed. You can see progress in completing each page. You could also outsource or delegate portions that you may not need to be the one to work on.

## Iterate Through Feedback

The longer you wait and try to figure everything out yourself or complete a whole project in one large swoop, you are putting time, money, and energy into creating a product that you haven't validated. You can guess at what features your customers may want, but it would be better to get real feedback from real customers.

If you create a minimal increment, you can show that to people and get feedback on what they like, what they may not like, and you can incorporate those changes to improve the quality of your product or service. The whole concept behind working in iterations is that you can better manage a smaller scope of work than trying to tackle a huge project all at once and complete an increment that you can get feedback on.

## Focus on Prioritization

Think about the program Microsoft Word. How many of those features do you use on a regular basis? Many people will say they use maybe between 5 to 10 percent and at the highest, I've heard around 20 percent. That means that 80 to 90 percent of the features and program capabilities are rarely or never used. Imagine how long it took to create all of these other robust features that weren't as important as the ones that are used most often (bold, underline, spellcheck, save file, save to pdf, etc.)

When you look at what you should focus your efforts on, look for your 80/20. What can you spend 20 percent of your effort on that will yield 80 percent of the results? Look for what is most valuable rather than trying to do everything. Look for what will bring the greatest return and start there.

When you try to build the perfect product or launch the perfect program, there are often challenges that can lead to feeling overwhelmed. Try focusing on a Minimum Viable Product instead. Find a minimal increment that you can release and get feedback on, then build on it from there. That's Agile.

The opinions expressed here by Inc.com columnists are their own, not those of Inc.com.

***

To read this article online with embedded hypertext links, go here:

**https://www.inc.com/maria-matarelli/use-this-1-method-to-do-more-work-in-less-time.html**

# About Maria Matarelli

**Maria** Matarelli is an international business consultant, experienced Agile Coach, and Certified Scrum Trainer (CST) who consults and trains companies on reaching true agility and getting results. Maria travels to consult Fortune 100 companies and speak at industry conferences with locations from Shanghai to Singapore and Thailand to Nova Scotia. While looking for ways to continue to expand Agile outside IT, Maria and her team have been applying Agile to the Marketing realm with incredible results, which led to Maria co-founding the Agile Marketing Academy. Maria also co-founded the Personal Agility Institute, helping people use Agile to do more of what matters in their lives and shift the culture of organizations. In addition, Maria is the Founder and President of Formula Ink, an international consulting company and is passionate about working with people and organizations to inspire agility.

# Do you know Kung Fu?

By Blake McMillan

Posted on July 26, 2018

When we say we know something, that can be true but at the same time there can be considerable differences in level of competency between those professing their knowledge. A student who is in their first few months of studying Kung Fu could claim to know Kung Fu. However, they wouldn't compare to someone like Bruce Lee who was not only an extremely skilled practitioner but also a teacher as well. They both know more about Kung Fu than someone who has never studied, but only one of them would be considered a Kung Fu Master.

One of the roles on a Scrum team is the Scrum Master. The word Master is included and applied to anyone who functions in this role. This implies mastery in Scrum in the same way Kung Fu Master implies a mastery of Kung Fu. However, the designation of mastery for a Scrum Master is misleading. When I attended a 2-day Certified Scrum Master class back in 2006…at the end of the class I was congratulated for becoming a Certified Scrum Master (that was before the assessment was required). Did I exponentially increase my knowledge of Scrum over those two days? Absolutely! Did the two days of training give me the knowledge I would need to claim mastery over the subject of Scrum? Not even close! I did my best to incorporate some Scrum concepts into how the team worked, but I did not continue to pursue knowledge and mastery of Scrum.  Many years later, when I decided to pursue Professional Scrum Master training through Scrum.org something different happened. Although I successfully passed my PSM I assessment, that was not the end….it was just the beginning. I kept studying Scrum using a variety of methods and you can see some of the books I read and recommend in the Resource section of soulofscrum.com. One of the discoveries I made as I learned and progressed: the more I learn…the more I realize how much more I must learn.

As Scrum Masters, it is important for us realize that we should be continually learning to best serve our teams and gain mastery of Scrum. Here are a few reasons pulled directly from the Scrum Guide about the role of the Scrum Master:

- The Scrum Master is responsible for promoting and supporting Scrum as defined in the Scrum Guide. Scrum Masters do this by helping everyone understand Scrum theory, practices, rules, and values.
- Ensuring the Product Owner knows how to arrange the Product Backlog to maximize value;
- Understanding and practicing agility
- Coaching the Development Team in organizational environments in which Scrum is not yet fully adopted and understood.
- Leading and coaching the organization in its Scrum adoption;
- Planning Scrum implementations within the organization;
- Helping employees and stakeholders understand and enact Scrum and empirical product development

To do those things, we need to become experts in understanding and applying Scrum. So, you might be asking what is the right way to keep learning continuously? I don't know…. that will depend on how you learn the best, where you live, and what your level of experience is. Here is a list of some suggestions to explore:

- Learn continuously by getting feedback from the team that you support. As you help your team learn and grow in Scrum, look at it as a gift when they provide feedback to you about how you could improve.
- If you haven't taken the Professional Scrum Master training from Scrum.org, consider it. Although there are

differences between instructors, the content is standardized and very useful for gaining a good understanding of Scrum.

- Read books…or if you are like me and you have a somewhat long commute listen to audio books. I've included some of my favorites at https://soulofscrum.org/resources
- Another great option for the commuters out there is listening to Agile/Scrum podcasts. I enjoy listening to the Scrum Master Toolbox Podcast with Vasco Duarte and hearing from different Scrum Masters around the world.
- Go to Scrum meet-ups. This is why I said it might depend on where you live. There are opportunities to meet with other Scrum Masters here in DFW. Improving in Plano TX does a good job of supporting those types of Scrum/Agile Community events.
- Talk to other Scrum Masters. It is great to share ideas and experiences and learn and grow with each other. I experienced that at Fenway Group w/ Brit and Tanner, at Toyota with the Scrum Master Quality Circle, and definitely with my fellow Improvers and the team at GMF.
- Look for opportunities to help or support other Scrum Masters in their endeavors.

Do you have other ideas of how to continuously improve and gain mastery of the Scrum Kung Fu we are called upon to know and practice as Scrum Masters? Please provide your feedback or comments if you see this in a social media format. Keep learning! Keep practicing! Keep Scrumming!

***

To read this article online with embedded hypertext links, go here:

**https://soulofscrum.com/blog/f/do-you-know-kung-fu**

# About Blake McMillan

**Blake** McMillian is a Principal Consultant at Improving with a focus on Agile/Scrum coaching and training. He currently holds Professional Scrum Master III and Professional Scrum Product Owner II certifications through Scrum.org and is working toward becoming a Professional Scrum Trainer (PST). With nearly 25 years of experience in the tech industry, Blake has supported companies such as GM Financial, Toyota, Capital One, American Airlines, Southwest Airlines, and ExxonMobil on a variety of projects.

Outside of working hours, Blake is passionate about sharing his knowledge and learning with others by presenting at conferences and events, creating and facilitating Agile Game sessions, developing content for the SoulofScrum.com website and blog, and serving as a Board Member and workshop facilitator for CareerSolutions (a non-profit that is dedicated to providing educational workshops for job seekers).

# Twenty Top Fails in Executive Agile Leadership

By Ian Mitchell

Posted on March 22, 2018

“Most executives, many scientists, and almost all business school graduates believe that if you analyze data, this will give you new ideas. Unfortunately, this belief is totally wrong. The mind can only see what it is prepared to see.” - Edward de Bono

If you are ever hired as an agile coach, it's quite likely that the importance of the engagement will be impressed upon you from the outset. That's certainly been my experience anyway. I've never yet been recruited for a gig which was described to me as being trivial. Why is this the case though? Are chief executives looking at their charts and numbers and saying "there's only one thing for it, we'll have to bring Ian Mitchell in"? Am I a high-rolling celebrity coach who only gets top-level contracts? It would be nice to imagine so. However my accountant at least, I am sure, has reason to suspect that this might not be true.

Rather, it's much more likely that when an organization is stressed, the pressures end up being applied to an external locus - namely the coach being hired to provide remedy. As a new and perhaps unknown quantity, all sorts of characteristics can be imagined onto a coach's blank dial before he or she arrives. Some may see you as a force to be channelled and directed against their cankered enemies, given that a powerful and mysterious magic will be heralded by your dawn. Get a good agile coach on board, it is often assumed, and new wonders will begin. Things will become faster and cheaper. The agile coach will bring change and make others see reason. Just watch!

Of course things don't quite pan out that way. Unless you have your own issues, you’ll recognize that no transformative magic can ever be radiated by your mere presence. You'll know that people have to want change, to realize that it affects them, and that sponsorship for engagement with a coach must be communicated and reinforced from the very top. Unfortunately, this can also present a significant problem. The difficulties may become apparent in the first meeting you have on a client site. To understand why, let's review the direction an initial rencontre avec eux might actually take.

“I hope you realize just how important this agile coaching initiative really is", you are informed with gravitas. "There's a lot riding on it. It's big, urgent, and critical to the future of the company.”

“Fine", you reply. "Let's get a meeting with the CEO into the diary. That’s who’ll have to sponsor the deep and pervasive organizational change which will be required.”

“Ah...um...well...”

You see, what often happens is that the supposed scale, importance and urgency of the challenge is passed onto the coach. The agile transformation initiative is handled like any other - through delegation. What isn't realized is that there's precious little an agile coach can do with exhortations such as this, no matter how gravely they are intoned. The coach does not have a hand-cranked transformation machine which can be turned any harder or faster. The constraints which inhibit change are much more likely to be found within the organization itself and across its serried departments, cultural assumptions, and established practices. That's why a good coach will value executive sponsorship and be keen to secure it. Anything less is likely to result in drift and delay, and the exposing of impediments which will stagnate unresolved. There may be local optimizations to scrabble for which can be worthwhile, and a window of transparency over critical issues and dependencies can certainly be put in place. At a strategic level though, the gains will be poor and it is unlikely that the high expectations will be met.

The simple fact is that responsibility for enterprise change cannot be delegated away to subordinates and hirelings. Senior executives are accountable for what happens, and they must be careful and keen to engage with any party who would help revise the model through which the organization delivers value. Yet so often they are conspicuous only by their absence, and by an unmet need for clear executive authority which only they can provide. It is their middle managers, and not coaches, who are served and who presently make decisions on what the disempowered should do. Ironically, these can be the people whose interests are most likely to be vested in the status quo. They've been there for years and know how to work the system and exploit its foibles. If a coach tries to make a connection with the senior higher-ups in order to explain and resolve an organizational impediment, mid-tier managers are likely to intervene and put a stop to the attempt or otherwise contain it. The organization’s officers act so that control of access-to-power is reasserted through the hierarchy. At times, hunting down a true executive sponsor for a supposed agile transformation can seem like an episode of the X-Files. As soon as a coach gets close to what might be the truth, the Men In Black appear, complete with smart suits, a threatening robotic manner, and dire warnings of how important it is to back off and take things no further.

The risks presented by weak executive sponsorship are great. A company does not typically stop in order to transform, and enterprise change must happen while an organization is in flight. Ultimately it is senior managers, and not the coach, who will be accountable for the success of the endeavor. So now let's examine some of the more common failure modes which executives demonstrate when abdicating their agile leadership responsibilities.

## Fail #1. Not recognizing the immediate significance of organizational culture

These days pretty much every chief executive talks about becoming agile. For the most part, they would genuinely like to think of themselves and their people as being able to work in this way. However, the executive view of their organization is cast through the lens of their immediate reports, filters, and barriers to access. This is the domain where the "frozen middle" hold sway. Any assessment of cultural impediments to agile practice will therefore pass through and be shaped by the system itself. The information they receive about the organization will be inherently self-referential and objectively blurred and distorted. Yet through this intelligence executives hope to redefine value streams and portfolios and roles, and to discern a strategic agile policy. They wish for a telescope to see further and are handed a milk bottle to look at the moon.

Nevertheless, the failure belongs with them. Everyone must recognize the significance of culture in defining what is seen now and the way it is seen. Agile transformation is indeed deep and pervasive, and understanding that it somehow "involves" cultural change is not enough. It's also essential to realize that the information you receive, and the way you receive it, may no longer be fit for purpose. Hence the way executives personally interact with their organization must change too.

## Fail #2. Thinking that agile change is technical

We've all seen it: the roll-your-own "agile" model which draws a loop around the development stage in a waterfall process. We cannot feign surprise. Software, being inherently plastic, is the point of least resistance when it comes to organizational change. It's also where value is most clearly added to any products or services being built. Other enterprise processes and artifacts are further removed from the empiricism provided by demonstrable working code. These other parts of the value stream often move glacially, and are abstracted away into more rarefied administrative layers. Yet all of those requirements sets, design specifications, sign-offs and promissory notes effectively define the established way-of-working. They are consequently much harder to challenge, and harder again to genuinely change.

This can easily mislead stakeholders, including senior executives, into thinking that agile transformation is essentially a technical concern. The result of this bias will be weak pull for any deliverables which those "agile development teams" are expected to create. Also, any dependencies they may have upon other organizational elements, such as other teams, architectural authorities or change control boards, will remain unresolved. There may be local optimizations and improved transparency over old problems, but any significant change to the way business is actually conducted cannot be expected to happen. It's incumbent upon senior executives to realize that if the benefits of agile change are indeed to accrue, then change must be demanded from - and sponsored across - the whole enterprise.

## Fail #3. Trying to delegate responsibility for change

Most organizations, especially large ones, are hierarchical. The way in which value and decisions flow across the enterprise is shaped by this structure. Broad decisions are made at the top. The details are left to middle-management layers to sort out, oversee, and implement. Progress will be gauged not through empirical evidence, but through the reports and pro-forma mechanisms expected by those delegates, and which transfer accumulated risk from one party to another.

This can lead executives into assuming a hands-off posture when seeking enterprise change. A decision may be made to "go agile", and perhaps to even adopt a particular framework, but the execution of the plan will be delegated. This posture may be reinforced by the specious argument that employees ought to be self-organizing in this new agile world. The duplicity of this logic is quite clear - the people can't self-organize because the culture for doing so hasn't arisen yet. Moreover, by delegating risk through and across a hierarchy, empirical control of the change process will be lost. How can an executive know that a plan is correct, and correctly understood, and correctly acted upon? People may see a need for change, and genuinely want it, but they rarely want to change. The responsibility for implementing a transformation is then delegated ever-further downward. It must happen elsewhere. Eventually it may land at the feet of an agile coach...someone whom few can spare time for, but from whom great results are expected. Again, this is a top-level executive failure. Everyone must be involved in enterprise transformation, and the responsibility for managing its success cannot be passed on to someone else.

## Fail #4. Not appreciating the effect of organizational gravity

Organizational gravity can be defined as the tendency of people to revert to traditional ways-of-working. This is often the case in an agile transformation initiative where traction is not gained. Employees may observe old procedures and protocols while emulating new agile practices as well. It can be seen as the state of doing agile without being agile. When a coach eventually leaves, not even lip-service to the new system might be paid any more. Organizational gravity can also cause people to behave defensively when faced with the expectation of genuine change on their turf. A company's "frozen middle" management layer can see it as their duty to stop a dangerous revolution from happening, while mollifying senior executives with carefully engineered optics that suggest change is indeed underway. The organizational antibodies can be expected to kick in when the status-quo is threatened. There can be a great reluctance to see existing roles undergo metamorphosis, or even just to hold previous reporting structures in abeyance.

Senior managers must understand the withering effect of organizational gravity, which tugs and tears at a transformation attempt until everything becomes molded to the organization's current shape. They must be particularly wary of agile vocabulary being co-opted, where new words are merely used to describe unchanged roles, artifacts, or events. They must be equally wary of that new vocabulary being modified to fit in with an organization's existing values and prejudices. The result would be a debasement of agile practice and a loss of inertia for achieving the new ways-of-working that are expected. Executives owe it to themselves and their company to be properly informed about the difficulty of change, and what the words describing change really mean.

## Fail #5. Not sponsoring change robustly

Not all senior executives are bluff and headstrong captains of industry. Some are weak and easily blown by the prevailing winds, or are too detached from the enterprise and its work to be truly bothered about a change initiative. They may have given the nod to an agile transformation attempt when pressured by certain factions, but they will leave any handling to their direct reports to sort out. This is different to and worse than delegation, since the expectation of a managed and co-ordinated result isn't even there.

The consequence of weak executive sponsorship is a piecemeal approach to change. Some change may occur in isolated pockets for brief periods, but the critical mass for enterprise transformation will never be gained. The strength of agile sponsorship is a critically important factor in overcoming the organisational gravity which holds change back.

## Fail #6. Not communicating the requisite sense of urgency

John Kotter's "8-Step Process to Leading Change" ought to be instrumental in guiding executives through a transformation attempt. Usually however it is not. Even the very first step - creating a sense of urgency for change - is rarely followed through in practice.

A well-communicated sense of urgency is arguably the purest expression of good agile sponsorship. Imagine the possibilities were it to be asserted at board-level that not only must change happen, it must happen now. Conversely a failure to transmit this imperative across the enterprise, and to set expectations accordingly, will stop critical momentum from building up. Change will be put off or delegated until the scale and immediacy of the challenge recedes from view or can be deposited at a new coach's feet. Senior managers must take care to ensure that agile transformation is given top priority, and that adopting and improving agile practice is not an afterthought but instead must anchor everyone's working day.

## Fail #7. Executing an agile change initiative as if it were just another program

The modus operandi of an enterprise defines not only the way in which it delivers value, but its very method for implementing change. The processes it follows describe how change is allowed to happen, whether it be the construction of a new product to be added to a portfolio, or a new system to replace an old one through which institutional data is manipulated and retained. The processes themselves are generally assumed to be immutable until they are challenged by transformative agile thinking. They can appear to be the only conceivable way of doing things...an inherited bias which corrupts the art of the possible. Its strictures can even masquerade within the organization as common sense.

The ramifications of a transformation attempt are clouded by this thinking, which lingers as a sort of original sin. It pollutes rationalization and militates against an objective understanding of its forces. Senior executives, following established doctrine, may therefore try to delegate an agile transformation initiative as if it were just another program of work. It will be budgeted for in the traditional manner, given a standard mandate, and its success measured using old-style metrics in an old-time way. In effect the agile coach is transmogrified into a known quantity whose remit is already circumscribed by establishment orthodoxy: a program manager. The ability to apply empiricism, and therefore to prove value, will also be lost. Obviously the success of a change attempt will be severely impacted by this, and most likely from the very beginning. Executives must appreciate that an agile transformation will fundamentally change the organization itself. The whole enterprise is likely to be affected even if one team is to succeed, and hence the initiative cannot be subject to established program constraints.

## Fail #8. Trying to ride an agile change initiative on the back of another program

A further degeneration of the spirit of change is regrettably common. Instead of being framed as a true enterprise initiative, it might be applied as a salve to another program of work, and perhaps one which would otherwise be dismissed as being hopelessly over-ambitious. Examples of this are the so-called "digital transformation" schemes which have become fashionable across government and the larger corporates. The scope of such a scheme can be one of large-scale migration and technical uplift, often with a view towards abandoning troubled legacy estate and a necropolis of technical debt. However, any "agile" part of the initiative is rarely thought through. A mystery to those concerned, agility is conjectured to be the secret sauce which will align time, scope, and budget.

The immediate consequence of this toxic exercise is to constrain sponsorship for change to the program concerned. Where teams have organizational dependencies outside of the host program, such as upon shared corporate resources or authorities, those impediments are not likely to be resolved. The remit for change will not extend into those areas, and there may be no obligation felt there to help agile teams meet their delivery responsibilities. Any work done will then fall short of the standard needed for it to be considered finished. The Definition of Done will soon be outweighed by the deficit for release once development starts. Work will be batched up, incomplete, until those who lie outside the program are in a position to deal with it.

Senior managers should recognize the danger of constraining agile change in such a way, and the folly of expecting program directors to provide organizational sponsorship beyond their narrow remit. No matter how sure it seems that all necessary resources lie under program control, as soon as a team starts sprinting unforeseen impediments and dependencies will surely emerge. Agile change is deep, pervasive, and must be expected to transcend all existing work and associated programs. There's nothing wrong in selecting a pilot scheme, but sponsorship for agile practice must reach beyond the narrow horizons which currently limit thought and action. It can only really come from the very top.

## Fail #9. Trying to hire “method installers”

I remember being on a large agile transformation attempt at a London bank, where I was one of many coaches. The bank insisted on having clear transformation backlogs in place for each would-be agile team, of which there were well over a hundred. Planning sessions, reviews, retrospectives, usage of artifacts, presence of roles and so on were enumerated in some detail.

Make no mistake about it, transformation backlogs can be a very useful thing. They can and should establish transparency over progress, so it can be inspected and adapted and correlated to improved value. Unfortunately that's where the client's approach fell short. Change wasn't empirically measurable, it was something to be "coached out" to existing workgroups. Those groups had neither the authority to self-organize as proper agile teams, nor the time nor the remit to assay deep structural change. A coach had to work agile improvements into and around these busy people with their various silos, competing demands, and ongoing commitments within an otherwise stage-gated institution. There was no correlation between anything on a transformation backlog and any value which might be delivered in a timely way. The feedback loop was not likely to be closed for nearly two years. Worse, there were notional targets for attaining levels of supposed agile maturity during that dark period. Yet if there is no empirical evidence of improvement, how can maturity or progress be gauged at all? The bank thought they had their answer: they would conduct their agile transformation by checklist. If a coach, through some mystical means of divination, figured that a change had been coached satisfactorily then it was checked off. There were burn-downs showing how splendidly everything was going as each transformation backlog was whittled away.

You see, in reality we were not engaged as agile coaches at all. Rather, the bank had set us up as "method installers", and in so doing they had set their transformation up to fail. No business can hope to reduce a profound change in its culture to a check-box exercise. Amongst other things the validated learning loop must certainly be closed frequently, if an assessment of progress is to be grounded in anything other than fiction. The temptation to do otherwise can be considerable, since establishing empirical control within large organizations is hard. The path of least resistance can appear to be to set up false controls, with metrics and measures that are poorly vested in empiricism, but rich in convincing detail. Any assurances received will be fake. Senior executives must take care to ensure that an agile change initiative, including any work enumerated on transformation backlogs, is rooted in the empiricism agile practice brings to bear.

## Fail #10. “Mapping” instead of changing

About twenty years ago, before agile practice had even become fashionable, other means of iterative-incremental delivery were nevertheless in the ascendant. I remember many companies spawning their own versions of the "Unified Process" and how I assisted in some truly monstrous births. It was thought that change could be eased into a hidebound organization by "mapping" its stage-gated process onto a new iterative and incremental model. Improvements in the flow of value would be catalyzed by this exercise...or so it was hoped. In practice change very rarely happened at all. It never seemed to develop beyond the boxes, lines, and other squiggles which constituted the relevant process map.

The underlying issue is one which organizations still face today...people want change, but they don't want to change. The head of steam for improvement might build up no further than the creation of that theoretical mapping between an idealized agile state and the current reality. Institutional apathy can be especially irksome when a bimodal IT strategy is propagated. The ability to "drop out" of agile practice can be seized on, not when it makes best sense for value delivery, but when agile change simply proves too hard. Plus ça change, plus c'est la même chose.

Senior executives must therefore be genuine in their desire to bring about agile change. No benefits can possibly accrue from a paper-based exercise in which an agile "mapping" is thought to hold transformative power.

## Fail #11. Trying to change agile practice instead of changing the organization

When an agile transformation initiative is announced in a company, employees can see a need to either reconcile or choose between two quite opposing forces. In the one corner we have lean-agile practice and its flyweight techniques for value delivery. In the other we have the super-heavyweight enterprise and its established way of doing things. It hardly seems a fair match to begin with. Moreover the existing system already applies to the organization in its own context. In comparison, an agile coach only seems to offer generalizations about certain practices which have been applied elsewhere. Who will listen to a dreamer like that in the ring? "Things are different here", it is very often claimed. "Vanilla agile won't work!" There is a perceived credibility gap to be bridged. Anything from introducing Sprint Goals to having a release-quality Definition of Done might be dismissed as being "too vanilla" or "too generic", if it doesn't quite square with the current set-up. The agile coach is at a clear disadvantage when the old guard of middle managers exercise a right of veto. If anything must change, they are likely to say, then it must be the agile practices you have in mind. An industrious coach is expected to provide something which has been "customized" or "configured" to align with the existing deal. The arc of the agile transformation is brought down by organizational gravity. "Bring us an agile way-of-working that matches what we already do", is the obvious yet unspoken implication. Anything less than that is seen as a failure to be practical. For many coaches it is tempting to acquiesce, collect the money, and live a quiet life until the rubes get wise about the scam they've bought into and it's time to move on.

Every executive must therefore understand and accept that "agile transformation" means changing the organization, and not changing agile practice. That is after all the purpose of the exercise if any benefits from change are to accrue. They must also be aware that a lucrative industry has emerged to satisfy the popular demand for half-baked transformation attempts which do not yield a satisfactory outcome.

## Fail #12. Misunderstanding value and flow

Senior managers tend to think of agility in terms of executing projects more quickly and more cheaply. That's the secret sauce they often hope for. In truth though, agile practice is really about gaining empirical process control in a complex and uncertain environment. Those iterative and incremental techniques are geared towards experimentation, and at times the achievement of profit can seem almost incidental to that concern. Agile practice helps an organization to learn about the business context in which it truly operates. Agility isn't about running projects faster and cheaper. It's about learning to build the right thing at the right time.

Executives can be suckered into assuming a "faster-and-cheaper" prescript because they misunderstand the significance of value and flow. They may suspect both are important in achieving lean efficiencies. However, they might easily ignore the importance of visualizing the flow of value so business operations can be better understood. Most critically, they can fail to challenge the received wisdom of what "value" amounts to in the first place. Ask a manager what "value" is, and there's a good chance that "doing projects faster and cheaper" will be the depth of their reasoning. Middle managers, under pressure to complete projects under constraints of time and budget, are most likely to be nonplussed by a pursued line of inquiry. "Of course projects must be done faster and cheaper", they will think. "That's always the pain-point. What more to value can there be?"

Senior executives must be aware of how this thinking shapes organizational culture. They should be very careful to avoid framing agile discussions in "faster and cheaper" terms lest this mislead others or confirm their prejudices. Rather, they should clearly highlight the need for validated learning, the pull for small and incremental experiments, transparency over queue and batch sizes, value prioritization techniques, and team self-organization around clearly defined workflows.

## Fail #13. Not thinking about what "Done" means

The degree to which a product is complete, and fit for purpose, is a common source of misunderstanding between stakeholders. For example if a product increment has been developed - but not tested under user conditions - then those working in an engineering department might still think of the fruits of their labor as being "complete". For them at least, it is finished. An end customer however might have different ideas. They could reasonably expect that user testing has indeed been carried out. They might additionally expect that documentation is in place and that training courses are available along with product support. So when is work truly finished?

Clearly there are implications here for quality control and for business reputation. Imagine what would happen if cars trundled off an assembly line without seat upholstery or headlight bulbs in place. In truth though, manufacturing industries generally have a robust understanding of what "done" means for their outputs to be production-ready. The IT industry is perhaps not so mature or circumspect in defining and assuring its quality standards. A "Definition of Done" therefore ought to be of critical importance to senior executives, because they ultimately carry the can for corporate risk and reputation. They need to be sure that the definition being subscribed to by development teams is appropriate for enterprise purposes and that it would not put the business in jeopardy. Managers should realize that the most effective way to check work for compliance is to use skilled inspectors at the time and place of work being carried out, and not as a separate stage or phase. Frequent and timely inspection stops waste from accumulating and reduces any rework needed. Automated checks reduce the time spent on inspection and the chances of error.

This also means ensuring that teams do in fact have all of the skills needed to create a truly "Done" increment, including all testing, documentation, and handover or support capabilities which might be needed. Being able to provide increments of work that are demonstrably complete, early and often, is instrumental in assuring transparency over progress, controlling risk, and minimizing the accrual of technical debt. Senior executive sponsorship may be required if any dependencies upon external resources, skills, or authorizations are to be eliminated and brought within a team.

## Fail #14. Not reconsidering metrics

In traditional organizations, managers often depend upon reports to give them an idea of what is going on and the decisions they ought to make. Charts and numbers provide them with essential information, and they interpret these sources using the indicators and measurements they think best reveal the underlying truth. Their view of the world is filtered and shaped by such metrics.

Unfortunately however, even transformational progress will often be gauged using these existing indicators. The time left until the supposed completion of the initiative can be thought most important for example, instead of the value which is being delivered now, and what might be learned from it. Executives should realize that the way success is measured is also something that has to change. More appropriate and empirical means of assessment can include a focus on innovation accounting and the elicitation of actionable rather than vanity metrics. Even measures of employee performance could have to be revised, so that better teamwork might be rewarded. Whatever techniques are chosen, executives need to accept that agile transformation cannot be sponsored or assured using old anachronistic measures and terms of reference.

## Fail #15. Not applying empirical process control to enterprise change

It would be foolish to try and conduct an agile transformation as if it were just another program of work. However, it would be equally foolish not to set about change in a managed way. Unfortunately though, a lack of control is apparent in many agile transformation attempts. Empiricism is typically lacking. Instead, a mixed slush of agile techniques will be thrown against the wall, in the hope that some of it might stick. It's a recipe for people to become adept at "doing" agile rather than "being" agile. They'll go through at least some of the forms and motions, for a while, but the empirical evidence provided by agile delivery is unlikely to be forthcoming. Organizational gravity is more likely to assert itself, and to bend change to the institution's current will and shape.

Empirical process control is inherent to any true agile way-of-working and that must include a transformation attempt. Hence senior executives must look for empirical evidence of improvement, and not reports or other promissory notes, to be assured that an agile change initiative is indeed "on course". For example, they might reasonably expect transformational effort to be visualized and managed as one or more change backlogs. Each should capture the good agile practices which are expected to "move the needle" and help achieve empirical improvement in team-based value delivery. At least one transformation group may be needed who can interpret the effect of agile coaching in this way, and inspect and adapt those transformation backlogs, until agile practice is normed across the enterprise and self-organizing teams are in place.

Managing a cultural shift of this nature is not something which can be delegated. There will be many things that need doing, and which will transcend any one program of work. They are likely to cut across the whole organizational structure. Even getting just one team to successfully deliver a release-quality increment after just one iteration can require deep and pervasive change across the enterprise. There will be many unforeseen dependencies which will have to be recognized, challenged and removed. New agile teams will have to be brought together with the ability to self-organize. There will be new value streams to be identified and old ones which will need to be improved. Teams will have to acquire new skills, to learn based on evidence, to sort out their dependencies with others, and to identify and remove waste.

If you were to get to the heart of what enterprise agility really means, you could say that it demonstrates the empirical mastery of innovation at scale. Since most organizations are conservative rather than innovative, executives must recognize the barriers which are likely to be presented to any associated agile change attempt. The organization is likely to harbor some quite reactionary forces, including middle-managers who genuinely believe that resisting change, and preserving established and "proven" norms, is the right thing to do. Very often, empiricism is something only resorted to in an emergency, and abandoned when the danger recedes. Senior executives must sponsor transformation across all of the parts of an enterprise, and advocate the empirical process control which always underpins agile practice.

## Fail #16. Not encouraging a product focus

Pick a large organization, any one. Go to the head office and take a good look. What business are they in?

Well, chances are you could be forgiven for thinking they're in the business of running projects. That's the main thrust of what they might seem to be doing anyhow...irrespective of whatever products or services they might provide or the industry sector they are in. They are likely to be drafting project initialization documents, seeking and approving mandates, scoping requirements sets, establishing project plans at various levels of detail, concocting RAID logs and RACI matrices, leveling resources, securing funding and putting together work packages. Everywhere, documents are passed around which are promissory notes for some value as yet unevidenced and undelivered. In a perverse expression of managerial self-aggrandizement, these documents may even be referred to as "products". Risk is transferred from one party to another until judgment day when the actual product is scheduled to materialize.

Projects are by definition temporary endeavors. Much effort is expended in setting them up, monitoring their supposed progress in the absence of regular useful deliverables, and in tearing them down afterwards. They are known to be a poor vehicle for sustaining the flow of value, and for retaining the institutional knowledge and team skills which allow value to be optimized.

Encouraging executives to rethink the enterprise not in terms of projects, but rather in terms of the genuine products which allow value to be released and empirical lessons to be learned, has proven to be a long and slow battle. Then again, developing a product focus might be criticized on the grounds that it doesn't necessarily emphasize value and flow. We could still have products which are ill-conceived and poorly represented in terms of ownership, and perhaps end up with limited visibility over services. Nevertheless, establishing a product-focused organization is undoubtedly a step forward to understanding value streams of any type, and it is a significant improvement on project-execution mode. Senior executives should therefore take care to ensure that products rather than "projects" are being sustained, and that they are clearly owned and represented at all levels.

## Fail #17. Not rethinking roles

Agile change is determined not just by the work that people do, but by how they begin to see themselves in relation to it. Any of an organization's present roles, perhaps even all of them, could need to change if value delivery is to be maximized. Senior executives themselves may need to rethink how they fit in to an enterprise value stream, and the value they should be adding. Is it really appropriate to expect "reports" from others, given that this can subject essential data to filtering and delay? How about building transparency, and even more importantly, making use of it? Furthermore, shouldn't responsibility for inspection and adaptation be devolved upon those who actually perform the work? Establishing transparency over the various things that are done, and the way people go about them, is essential to agile development.

In Scrum there are a mere three roles and Scaled Professional Scrum adds to this just one more. They are optimized to deliver work of release quality at least once every month. In comparison to this, large organizations can have scores or even hundreds of job descriptions. Each of them will be geared towards the scheduling of work according to the availability of the associated "resource", such as analysts and designers, programmers and testers. Work will be passed from one person, or pseudo-team with similarly constrained skills, to another.

Executives must understand that in agile practice work is not scheduled by resource availability, or by the passing of work between silos and through stage gates. Instead, people ought to go to where the work is, so that a so-called "resource" and the role they fill becomes orthogonal to the workflow. That's the intent behind a self-organizing team, and the teamwork they should exhibit. People should be able to organize themselves around the work that needs doing, when it needs doing, and ensure that it is done to a satisfactory level of quality.

## Fail #18. Overlooking the strategic, operational or tactical aspects of change

I've often found that when I'm hired as an agile coach, little thought is given to the type of coaching I will need to do, or how it is likely to impact the organization and its stakeholders. Sometimes it's just assumed that I will ensure an initiative is somehow accomplished "faster and cheaper". The classic error of executing agile change as a project or program, or on the back of another one, is rarely far away. More remote unfortunately is any consideration of empirical process control or of what it means to have a learning organization. The agile coach is simply thought of as another manager, albeit one who brings that “agile secret sauce"...the magic ingredient of project success. What executives fail to realize is that agile coaching impacts the whole organization, and hence it must be considered at tactical, operational, and strategic levels.

Tactical coaching is most often team-facing and innately unpredictable. There are always retrospectives or other events which need facilitating, or during which a Scrum Master could need to be mentored. There might be on-the-spot advice which is asked for or which ought to be given, estimation or other agile techniques which must be taught, and outreach sessions to other parts of the enterprise which need to be arranged. Tactical coaching is usually ad-hoc, and isn't something that an agile coach ever really escapes from, or should wish to. It's about helping people to understand and master agile practice in the field.

Operational coaching is comparatively focused, and aligned towards the specific agile roles people need to fulfil. Product Owners, Scrum Masters, and Development Team members can need formal or semi-formal training so they at least broadly understand the part they have to play, and the collaborative relationship they will need to have with others. Scrum courses which articulate to some type of accreditation can provide a certain degree of grounding. Also, there are people in existing roles, such as business analysts, testers, or project managers, who can feel uncertain or threatened by the prospect of what agile change will mean for them. Sometimes workshops can help people to move into new roles for which they are qualified. Business analysts, for example, may benefit from group sessions in which user story writing or backlog refinement are explained. In short, people may need to be coached to perform their agile roles at an operational level.

Strategic coaching is aimed at executives. The assistance given can be structured or unstructured, depending upon when it occurs in the engagement and which management functions are addressed. There might be a formal workshop or training course to begin with during which essential terms of reference are established. Later on, there might be more tightly-focused sessions during which many questions will need to be answered and risks clearly highlighted. Some of the issues and pitfalls covered here might be explored, for example. It's also vitally important to ensure that a strategic vision for enterprise agile transformation is elicited, and that adequate sponsorship for it is in place.

## Fail #19. Thinking they’re above it all

Sponsorship for agile change, unfortunately, very often proves to be the Achilles' heel of a transformation scheme. Part of the problem lies in a delegation culture and in assuming that the challenge is essentially a technical one. From this executives can compound their error by failing to consider the agile coach as an enterprise change agent who is deserving of their time. Rather, the coach is seen as a junior manager or technician, someone who works with developers and other technical-types so the agile gubbins is plugged in. There's little chance of being invited to the top floor, or the board-room, to discuss over coffee the various organizational impediments which you see building up. There's even less chance of executives coming down to where the work is done, and taking a proper look for themselves. The CEO's presence is reserved for management consultants on ten times your day rate, or real "celebrity coaches" with their kumbaya games and mood-music. You meanwhile are someone to be kept below stairs, under the control of the delegated subordinates who hired you for their project or program.

This is the most critical executive fail, since it drives many of the others we have explored here. It enforces a disjuncture between the vision of an agile enterprise, and the pro-active sponsorship which is needed to achieve it.

## Fail #20. Not changing themselves

If we were to look for a key take-away from this score of executive agile fails, it would have to be this. It's the reluctance of senior managers to change themselves. If you are one, try asking yourself these questions. Are you prepared to leave your desk, to wean yourself off reports, and to see for yourself what is really happening in your company? Are you willing to make agile transformation your top priority? Do you really want enterprise change, and not some sort of bolt-on capability which you think delegates ought to handle? Are you willing to challenge, and be challenged on, the very idea of what value means to the organization? Do you see the importance of flow, and the futility of making projections without empirical evidence?

Now think of it this way. There are many human behaviors we can wish were altered so our lives would be improved. However there is only one person whose conduct we can ever really change, and the agile transformation in your organization quite simply has to start there. I'm not going to spell out who that person is. And remember: agile transformation is a harsh mirror. If you don't want to know how ugly you are, don't look in.

***

To read this article online with embedded hypertext links, go here:

**https://www.scrum.org/resources/blog/twenty-top-fails-executive-agile-leadership**

# Monte Carlo forecasting in Scrum

By Ian Mitchell

Posted on September 17, 2018

"The essence of the beautiful is unity in variety."

- Mendelssohn

## My gears aflame

I remember being told, many years ago when I started university, that Information Technology is a numerate discipline. I had been admitted onto the IT course on the strength of a high-school biology certification. That was my "technical competency". The professors assured me, however, that I did not need to be an expert mathematician. The essential requirement, they said, was just to be able to work with numbers.

I took that to mean I needed to be able to add and subtract, which I had mastered at the age of six or seven, or perhaps a couple of years later when I suppose I must have been about twelve. Anyway, it turned out that they *really* meant I had no obligation to study mathematics beyond second order calculus. You could smell my poor gears burning as I hurtled into Remedial Class City. It can be argued that I have effectively remained there, trying to deduce inflection points on my fingers and toes, ever since.

I am the least numerate person you will ever meet in the IT industry. The people working in the office canteen are more mathematical than I am. At least they can work out the change. For my part though, numbers induce within me an absurd feeling of dread. They are at once both my nemesis and my nightmare. They have left me with my very own case of *imposter syndrome*. When will my arithmetic incompetence be exposed to all, I chew my nails wondering? When will I finally be caught out?

## Ducking and diving, bobbing and weaving

Agile coaching can be a good hiding place for someone like me, at least until metrics are discussed. Fortunately, it's possible to kick the can down the road for a while before that actually happens. An agile coach might assert, with justification, that true value lies in the product increment and not in throughput, velocity, or other proxy measures. Sprint capacity planning, you can also argue, is never meant to be a sufficiently exact science for which a mathematical approach might truly apply. You can rightly claim that estimates are just that – estimates - and that a team merely needs to get its arms around how much work it thinks it can take on. You might also assert that "agile team members should take care not to become story point *accountants*" - spitting out that last word as I do, like a hardcore numerophobe.

Sometimes though, an agile coach really doeshave to look at the numbers. If a Product Owner wishes to make evidence-based future delivery projections, for example, then it will be hard to help without conveying an understanding of burn rates, capacity, or throughput. Then there are times when a quantitative approach is arguably less essential, and yet you know in your heart it's the best option. Flow optimization during a Sprint can demand the analysis of cycle times for example, and not just the eye-balling of a team board for bottlenecks. Improving service level expectations can mean that work-item age ought to be assessed, since there are times when a general reflection on team performance just won't be enough to cut it.

## Crunch time

Sprint Planning is another occasion when an irredeemably scattered brain, delinquent in mathematical measure, can be forced into jackboots and expected to crunch numbers. It's a formal opportunity to use the best data available in order to provide the best possible time-boxed forecast of work. According to the available historical data, how much work can a team reasonably expect to undertake during a Sprint? It's time for you to challenge any numeracy demons you may have.

If you step up gamely to the mark, you'll bend your mind around a product backlog's size and the rate at which a team burns through it. A forecast of Sprint capacity and projected delivery dates will hopefully lurch into view. Pushing the envelope further, you might also factor in the possibility of a Sprint Backlog growing during the iteration. You elicit a burn-up chart to accommodate such eventualities, and then cheekily express disdain for mere burn-downs. Apart from an occasional dalliance with cumulative flow and Little's Law, that's about as far as a mathematical derelict such as I will generally dare to venture into the arcane world of numbers.

Fortunately, I have been able to comfort myself for years with the knowledge that, however incompetent I might be in terms of mathematical ability, most of my peers will demonstrate no higher accomplishment. While they must surely be better at grasping metrics than I am, for some reason they always evidence the same *dilettante* standard of burn-ups and burn-downs. With an ongoing sigh of relief, I have found again and again, that I can wing my way past the whole issue. Nobody seems to care about fancier metrics than those I can actually cope with. The bar has not been set high and even I might hope to clear it.

## Raising the bar

The problem is, these usual measures will typically express a precision with limited respect for accuracy. If we have to forecast when a delivery is likely to be made, for example, then we may offer a particular number as our answer. From the Product Backlog burn rate, we might project that the relevant increment will be forthcoming in "Sprint 12". The calculation we have performed will demonstrate that our assertion is fair. Yet we will also know that the number we offer is uncertain, and that the increment might not be provided in Sprint 12 at all.

We can usually express greater confidence in vaguer terms, such as a range. It might be wiser to say "delivery will probably happen between Sprints 11 and 13". You see, the more precise forecast of "Sprint 12" is not very accurate. It is a rough number precisely stated. We offer it as cipher for a vaguer and truer answer. This is a game we play with our stakeholders. We know, and expect them to know, that any number we forecast may not be accurate. It is an estimate.

Why don't we offer the sort of range we can have better confidence in, whenever we have to make a forecast? Why are we obsessed with giving a precise number no matter how unreliable it may be?

Part of the answer lies in our cultural reluctance to be vague, even under conditions of high uncertainty. We assume that someone who fails to be exact cannot be on top of their game. The charlatan who deals in fantasy numbers with precision is respected, while the careful analyst whose stated range actually proves reliable is held to be of lesser accomplishment.

Fortunately, there is a way out of this state of affairs. We can raise the bar and still hope to meet it. All we need to do is to ensure that any "vagueness" we express – such as a range of dates - accurately captures the uncertainty we are dealing with. If we can evidence that velocity is stable to within plus or minus 20% every Sprint for example, then the range we offer in a future prediction ought to reflect that established variation. In truth, the range we give will not be "vague" at all. It will be a range precisely articulated, and it will be as accurate as it is founded on hard data and careful analysis.

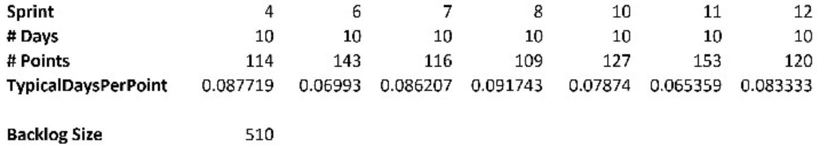
## An example

Let's go through a worked example. This is entirely from real-life by the way, and reflects an actual situation where I joined a team as a Scrum Master. All of the data is real and nothing has been made up.

The situation is that the team has been Sprinting for a while with iterations two weeks in length. There is a Product Backlog of an estimated 510 points remaining. We know the team velocities for Sprints 4, 6, 7, 8, 10, 11, and 12. These were 114, 143, 116, 109, 127, 153, and 120 points respectively. We don't have the data for any other Sprints and some Sprints are evidently missing. It doesn't really matter though. We just need a representative set of recent velocities from which variation can be evidenced.

We can see that the distribution falls between 109 and 153 points and with no obvious pattern to the scattering. We could of course work out the average burn rate, which is 126 points, and then estimate that it would take just over 4 Sprints to complete the 510 points of work. However, as we have already discussed, when we take an average we are effectively throwing away the variation in order to arrive at our answer, and so we lack an understanding of its accuracy. When we average stuff out, it should be because the variation is annoying and we genuinely want to throw it away. With our greater ambition however, we want to make use of any variation so as to deduce a more useful and reliable range.

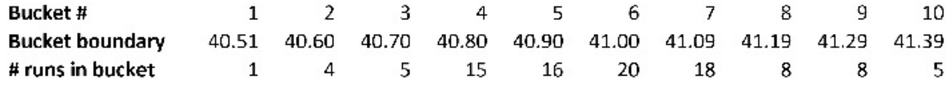
The first thing to do is to express the data we have in a more flexible way. Each Sprint lasted 10 working days. If a Sprint completed X story points of work, that means a typical point for that Sprint will have taken 10/X days to clear. We can now produce the following table from the data we have.



| Sprint | 4 | 6 | 7 | 8 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|
| # Days | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| # Points | 114 | 143 | 116 | 109 | 127 | 153 | 120 |
| TypicalDaysPerPoint | 0.087719 | 0.06993 | 0.086207 | 0.091743 | 0.07874 | 0.065359 | 0.083333 |
| | | | | | | | |
| Backlog Size | 510 | | | | | | |

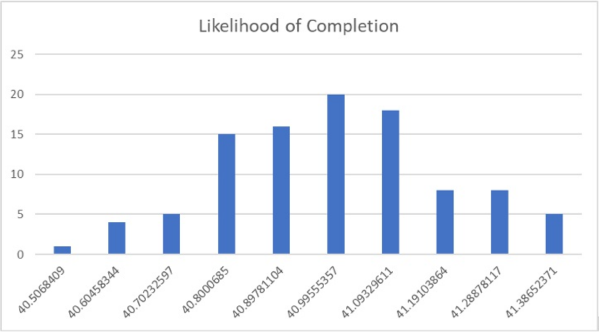
If you were asked to build a simulation of how long the 510-point backlog might take to complete, you could use the above data to help you. You'd pick one of the "Typical Days Per Point" values at random 510 times, and then add the set of numbers all up. If you ran the simulation once, it might come to 40.91 days. If you ran that simulation twice more, you could get totals of 40.63 and 41.35 days. On each occasion, you would be running a realistic simulation of how the team might be expected to saw away at 510 story points of work.

If you ran the simulation a hundred times, you'd get even more data to look at. However, the volume of data could then prove a bit overwhelming. So, let's group the run-times we get from a hundred simulations into 10 buckets, each covering one-tenth of the time between the fastest and slowest. In other words, we'll count the number of runs which fall within each of the equally-sized time-boundaries of ten buckets.

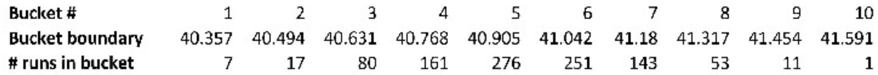


| Bucket # | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Bucket boundary | 40.51 | 40.60 | 40.70 | 40.80 | 40.90 | 41.00 | 41.09 | 41.19 | 41.29 | 41.39 |
| # runs in bucket | 1 | 4 | 5 | 15 | 16 | 20 | 18 | 8 | 8 | 5 |

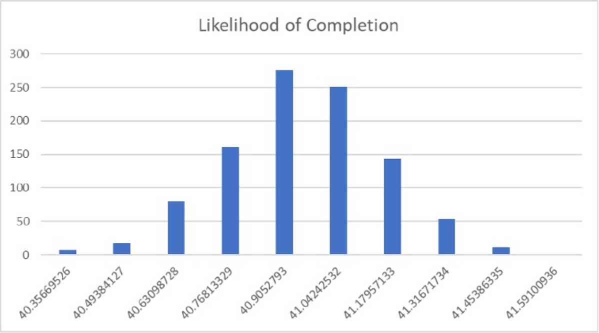
OK, all of the runs add up to 100. What we appear to have though is a concentration of runs in buckets 4, 5, 6, and 7, with a peak in bucket 6 and a tail-off towards the first and last buckets. In other words, the 510-point backlog is more likely to complete in the 40.8 to 41.09 day-range than at any other point in time. Let's chart this to better visualize the results.



Now let's run a thousand simulations, and see if this pattern can be confirmed and if it emerges even more clearly.



| Bucket # | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Bucket boundary | 40.357 | 40.494 | 40.631 | 40.768 | 40.905 | 41.042 | 41.18 | 41.317 | 41.454 | 41.591 |
| # runs in bucket | 7 | 17 | 80 | 161 | 276 | 251 | 143 | 53 | 11 | 1 |



By Jiminy, so it does. There really is a clustering. In fact, it's the sort of thing a biologist might recognize as a normal distribution, or "bell curve". This type of curve is encountered frequently when studying population distributions, height or biomass variation between individuals, pollutant concentrations, or other events which - when examined in the small - might appear to be without pattern.

This means that if stakeholders wish to know when a given item occupying a certain position on the backlog is likely to be completed, we can do better than to calculate a time based upon a potentially misleading average. Instead, we can show them the time-range in which delivery is likely to happen, and the level of confidence we would have in delivery occurring within it.

Most occurrences cluster around the average - but few if any will actually be the average. Very few people are of exactly average height, for example. Remember that an "average" can be stated precisely but rarely proves to be accurate.

Note also that if we were measuring throughput rather than velocity - or had a rule in which each story must be scoped to one point - then 10/X would effectively represent the "takt time" for that Sprint. This can be a better alternative to story point estimation in so far as it takes a team closer to gauging the actuals of stories completed.

## End note

We can never guarantee the certainty of a forecast. The scope of work can grow, items can be re-prioritized, and unforeseen events can always happen. True value will always lie in the increment a team delivers, and not in story points.

Nevertheless, when we collect metrics, such as the velocities attained over a number of sprints, we ought to remember that critically important detail can exist in the variation. All too often we just work out a forecast such as an average burn-rate, and throw the texture of the available data away. Yet in truth, we don't have to give stakeholders a single "precise" date for delivery in which we can express little confidence. We can do better, and show them the projected range from a thousand or more simulated scenarios.

The process we have covered here is sometimes referred to as a "Monte Carlo" method. This is a class of algorithms which use large-scale random sampling to generate reliable predictions. The technique was implemented computationally by Fermi, Von Neumann, and other physicists at the Los Alamos laboratory in the 1940's. They were working on problems such as how to work out the likely penetration of neutrons into radiation shielding. It's generally applicable to complex phenomena where the inputs are uncertain, and yet the probability of outcomes can be determined.

I suppose it's possible that the world's great nuclear physicists aren't much good at working out their change in the canteen either. What they are certainly very good at, however, is recognizing how new information – such as a reliable forecast - can be inferred or deducted from observations which others ignore or take for granted. That's a very special kind of skill.

As a mathematical blockhead, I can't lay claim to either of these abilities. What I can do though is to stretch a little. I can raise the bar a touch, and perhaps hope to meet it. You can too. We aren't restricted to working out an average velocity which we can plug in to burn-rate projections, for example. Monte Carlo analysis is just one illustration of why we should take care before stripping the texture of our data away. We should always seek to provide stakeholders with better information, using the data we have, and by the considered inspection and adaptation of our method.

***

To read this article online with embedded hypertext links, go here:

**https://www.scrum.org/resources/blog/monte-carlo-forecasting-scrum**

# Antipattern of the Month: Unlimited WIP

By Ian Mitchell

Posted on November 6, 2018

**Lean and Agile practices both explicitly prohibit taking on more than you can handle, so what fuels teams to continuously take on too much WIP?**

Lean and Agile practice are evidence-based and rely on the empirical delivery of value to assure predictability. Planning, development, and incremental release ought to occur on a short enough timescale, or cadence, to facilitate the inspection and adaptation of both product and process.

The even and consistent flow of work underpins this ability. Flow is enabled by a clear demand for completed work, the "pull" being propagated as efficiently as a team can manage across all of the stations at which value is added. Limiting "Work in Progress" helps to achieve this flow, and it is a very important skill to master. It ensures that work must be completed before anything new can be started. Each station should be given some sort of limit, such that anything below the limit indicates a capacity for new work to be brought forward. This helps to build pull across the value stream.

Limited WIP reduces lead time, cost of delay, and the depreciation of work has been commenced but not finished. Problems can arise when WIP is not subject to a clear limit. New work may be pushed or forced onto a team in an attempt to drive it through their engineering process, a common situation in a reactive environment where product ownership and backlog management is weak. Furthermore, team members may be keen to show they are "busy" and that they have a lot of work on their plate. This antipattern can require deep cultural change before it is resolved and flow is enabled.

A team with unlimited WIP may exhibit the following characteristics and constraints:

- Not everything in progress might be underway. The transparency over the work people are genuinely doing may, therefore, be poor.
- Much of the work that is thought to be in progress may, in fact, be blocked due to external dependencies, such as a need for certain skills or authorizations which the team does not have in order for work to be completed.
- Some of the work being performed may not be aligned to the Sprint Goal.
- The team boundary may be poorly defined or unstable, such that it is unclear who is committed and who isn't.
- The items in progress may not have clear acceptance criteria, and the point at which they are considered to be "Done" may be indeterminate.
- The items in progress may not be independently valuable and effectively constitute a batch.

## Unlimited WIP

**Intent**: Start new work before current work is completed

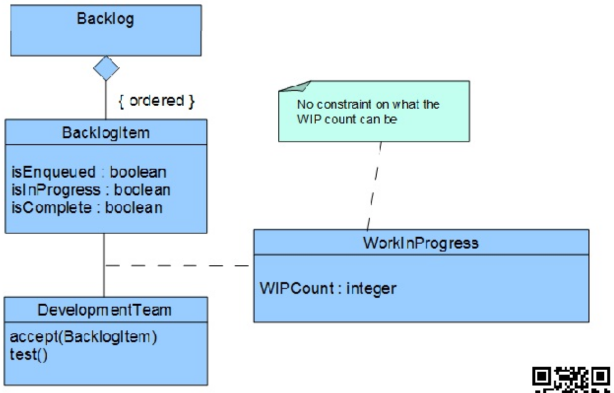
**Proverbs**: "More haste, less speed."

**Also Known As**: Unbounded Commitment

**Motivation**: When faced by competing and vociferous stakeholder demands, team members can feel obliged to action multiple items simultaneously. In an attempt to pacify stakeholders, they can, therefore, claim that an item is being worked on and is no longer enqueued.

**Structure**: A development team accepts items that have been enqueued in an ordered backlog. There is no limit to the number of items that are allowed to become a work in progress.

Unlimited WIP



**Applicability**: Unlimited WIP is common in reactive environments where product ownership is weak and/or scattered, such that no effective backlog prioritization occurs.

**Consequences**: Having no limit on inventory increases batch sizes and reduces the prospects of incremental release. Work depreciates while it remains in progress, and the timely delivery of value is compromised.

**Implementation**: Lean Kanban methods explicitly limit Work in Progress. Note that WIP limits may be applied to each Kanban states where work may become enqueued. Scrum does not strictly require WIP limits to be enforced as batch sizes are constrained in the form of Sprint Backlogs. Unlimited WIP is discouraged in all agile methods.

Youtube Video: The Unlimited WIP Antipattern

***

To read this article online with embedded hypertext links, go here:

**https://dzone.com/articles/antipattern-of-the-month-unlimited-wip**

# About Ian Mitchell

**Ian** Mitchell's experience in iterative and incremental delivery began with a PhD in object-oriented rapid prototyping in 1997. He then worked as a developer for several years where he witnessed the failure of many initiatives that were allegedly implementing Scrum. He realized that his true vocation was to help teams and organizations to better understand agile practice. Dr. Mitchell has a particular focus on evidence-based enterprise transformation and is the curator of a related site, agilepatterns.org. A Scrum advocate and accredited Professional Scrum Trainer, he is a veteran member of the scrum.org forums.

# The Attraction of Classic Hierarchies

By Kurt Nielsen

Posted on September 9, 2018

During the heat-wave over Europe this past summer, I realized that the little grey cells of our brains seem to suffocate and become slow and lazy in such conditions. It is simply hard to think seriously about something or design something new. Well, strictly speaking, I can only speak for myself, but one of the most obvious cognitive biases is that we all think everybody is like me.

Now that the weather is cooling down, I have gone back to some of the reflections on leadership and management that really puzzle me.

Before the summer we had some very intense training courses and presentations in Norway about Scrum and Agile Lean Leadership. There we saw many committed people, who really wanted to develop and learn. You have got to hand it to our brethren up there, they are not shying away from engagement and getting to the bottom of things. One question came back, over and over again in different settings, companies and organizations:

> *Why is it that we, who are in the front-line of creating solutions and servicing our customers and beneficiaries, want to work the agile and lean way, and our top management want the organization to be agile and lean. But those in the middle keep disappearing, stalling or disengaging?*

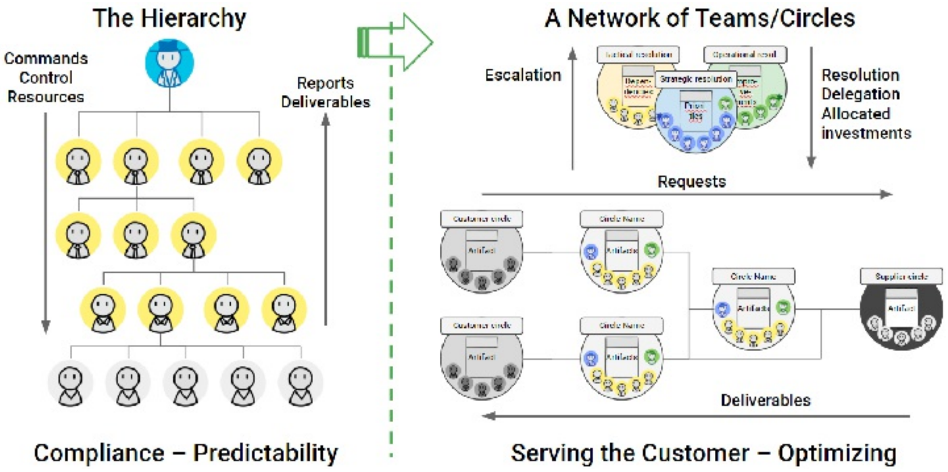
One discussion partner actually referred to the layers of middle management as “the permafrost”, he gave up and left. Having experienced this a number of times myself in real organizations as well, I began to look for some cross-cutting answer to this resistance from the traditional hierarchy in organizations.

I have spoken to many different people and each and everyone of them has added a piece to the puzzle. But whereas I thought there would be a fairly straightforward answer that could be expressed in a few sentences, I discovered that there are multitudes of little observations and inclinations that come into play in different combinations in different people; once again we are back in the complex domain.

I consequently set out to try and collect a list of the most important reasons and also some hints at what can be done to resolve the deadlock that many organizations apparently are in. This article is only an appetizer, I will try to deal with the individual reasons separately in later articles. I don’t want to be going over the same territory as in the original article about the pervasiveness of Neo Taylorism (read it here…). The focus here is on the resistance to Agile Lean Practices in the classic hierarchy.

## Prelude

First let us state the issue by looking at the following drawing. We are trying to persuade people and organizations to move across the green line to a more effective solution for complex work.



## The Hierarchy

To the left we have the classic org chart, with the executive, the CEO at the top, some layers of managers and experts and some grey matter at the bottom, that does the real work.

The system was originally conceived to ensure compliance, obedience, stability and predictability. In itself not bad or evil, but clearly not suited for periods of accelerated change or maneuver warfare. The purpose was and is the opposite.

Commands and resources ripple down, performance is monitored, control is applied and deliverables and reports bubble up.

Especially in our Scandinavian countries with low power distance (See Hofstede here…), the hierarchy is not normally brute force, but clothed in more modern, softer psychological clothes, such as in New Public Management. But the principles remain the same, just a different user-interface.

## The Network

To the right we have the proposed Agile Lean Leadership model of cooperating self-organizing Circles with teams.

The system models individual small operations held together by common values and goals. The key principle is organizing after the value stream keeping the eye on those we serve. There are of course decisions that are taken above the individual Circle level, we are not utopian here. But they are typically resolutions, delegations and investments decided upon by appropriate level 2 circles consisting of representatives from the primary circles. We call that the "New Structure".

There is of course a lot more detail to a complete organization, and we deal with that in the training program and our upcoming book: "Navigating the rapids".

## The Problem

While the blue guy and the grey ones on the left might be convinced that moving across the dotted line is a good idea, the yellow ones – those in the middle – don't seem to buy into it easily.

Very often people will say, "yes of course, these guys stand to lose in such a change, because the hierarchy is flatter, there are no middle managers". Often it is implied that the mere fact that we are not talking "up and down" is a loss of something. There seem to exist almost a dogma that says: what matters is how many you can yell "jump" at who actually have to "jump". That the position in the hierarchy is the basis of a person's self esteem.

What is true is that the yellow people will probably experience the greater change in their work, and certain class of work – the work of the administrator, will shrink.

I would argue the case that that is a good thing also for the men in yellow, but then I don't claim to be objective on that matter.

# The Sources of the Problem

Most of the causes mentioned below are not independent, they are highly correlated, and need to be viewed all together.

## The Neo-Taylorist Causes

These are some of the more obvious reasons for the resistance to the New Structure linked with the Neo-Taylorist view of organization. A lot of it was dealt with in our seminal article here…, so I will be brief:

1. The model of the hierarchy is a world-view, a paradigm. People cannot imagine that there can be anything else. They have never seen such a non-hierarchical organization, so clearly they will distrust it. However, I believe they have seen it. In society most people have given up on organizing according to Marx and Lenin with everything centrally planned. In private life we know that life cannot be centrally planned for very long. Somehow people still believe this is the only way in organizations.
2. Separation of thinking and doing. This is a serious one, it carries with it all the ineffectiveness, inadaptability, non-resilience and dehumanizing traits of assembly line production. It often also carries with it the conviction that thinking and planning is for an upper class of people and doing for the lower, it can give rise to a new sensation of superiority or aristocracy.
3. As a corollary of the point above, it breeds competitiveness for position in the hierarchy.
4. Another corollary is, that it is generally considered to be better to be a manager than being a doer, and considerable extrinsic motivation pushes in that direction.
5. The superior then becomes the customer, not the real customer. Because the superior is the one that controls the key motivator: moving up in the hierarchy.
6. The work in the hierarchy, pushing commands down and reports up, monitoring in detail etc. takes on its own life. It seems to become its own purpose, and it breeds more of the same stuff. Of course Parkinson discovered that already in 1955, read more here...

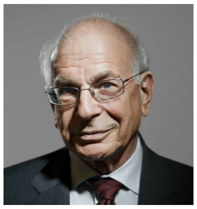
# Other Causes Uncovered

## Risk Adversity

The traditional hierarchy is built on control, you must appear to be in control and predictable to protect your position. This is drilled into managers. Therefore any proposed system or structure that seems to display less (albeit often imagined) control is threatening.

Daniel Kahneman (a subject for another article) studied this for many years. He says the following about another similar challenge (the use of uncertainty in decision making):

> *Decision makers don't like decision analysis because it is based on that idea that decision making is a choice between gambles. I think decision makers, in business and elsewhere, just reject the metaphor altogether. Managers think of themselves as captains of a ship on a stormy sea. Risk for them is danger, but they are fighting it, very controlled. The idea that you are gambling is an admission that at a certain point you have lost control, and you have no control beyond a certain point. This is abhorrent to decision managers; they reject that. And that's why they reject decision analysis.  Daniel Kahneman in strategy+business (2003)*



When you have lived for years in an environment where control and predictability is valued to the point of people faking the numbers to make plans and results look good, then it is very hard to see the world otherwise. As W. Edwards Deming said:

> *If there is fear in an organization, the numbers will be cooked!*

A hierarchy will much more often say no to initiatives than yes. By saying no, nothing happens and life goes on. By saying yes, you run the risk of something, not delivering the desired and planned results and you being blamed. So approving an initiative is a game with very little opportunity for winning and a lot of opportunity for losing.

And even more, Kahneman shows that even in the case of very little risk of failure, our human hearts interpret this irrationally as a much bigger risk, out of proportion with its real value. And that affects the decision making. “Better be on the safe side and not do anything!”

The solution to this is that top management must communicate a willingness to accept experiments with occasional undesirable results as a good thing that brings knowledge – sensible experiments of course, not any half-witted shot in the dark.

## Perceived Position and Respect

I believe much of this resistance revolves around the fact that managers in the Hierarchy feel that they will be less respected, fulfilled or appreciated in the new structure.

### Loss of Power

Power over other people is unfortunately attractive, it is not something to be proud of, because it only thrives on other people’s fear.

But having power over other people is attractive because it bolsters our self esteem and gives us something to show off in the competitive culture of many organizations.

In the new structure there is less ”power over” people, but more “power to” get things done and to influence people through skills, character and passion.

### Aristocracy

Being a manager can be a bit like being a member of the aristocracy of old. You are somewhat above the work and in some cultures it is actually a sign of high position that you don’t work, something that commands respect.

It is a class-based thinking that is not very nice to admit to, but we all do it in some circumstances; we lift our opinion of ourselves by defining us as different from these other people. Aristocracies always create closed circles to protect privileges and later on preserve the bloodline.

The managerial position with its air of superiority seems to be a privilege that has to be sacrificed if the New Structure is implemented.

### Fear of Breaking Rank

If you belong to a closely knitted group, a lot of your self understanding hinges on you position in this group. It takes a lot of courage to challenge the core beliefs of one’s core social group and potentially end up in bad standing. It is therefore connected with considerable risk for a manager to contemplate alternative beliefs about the nature of management or leadership.

## Extrinsic Motivation

The traditional hierarchy normally carries with it hefty increases in compensation and other perks as one moves up. As we know from the discussions about intrinsic motivation (see Anders Dysvik, New Scientist 9th April 2011 pp 40-43), application of extrinsic motivation overrules the intrinsic ones.

So if you have a motivated person and make him a manager and set him on a hierarchical career path ("You either move up or you move out!", like in the US Army), then by definition you have disabled or at least subdued his intrinsic motivation.

We know that people don't want to give such privileges up. It is not that managers are bad people, we would all be like that in such situations. And if the hierarchy offers you that, it is much easier to decide to stick with it.

Ever since the 1987 movie "Wall Street" with Michael Douglas as Gordon Gekko, his famous statement "Greed is good" has been more and more accepted as a fact. It is largely accepted and OK to be greedy and self serving, it didn't use to be like that. The acceptance of greed as a driver has been highly amplified by the media constantly focusing on people with extraordinary spending power. Most of us rarely see any other role models in the media, and it creeps under the door as a fog: "Greed must be good!"

There may be a spot of greed driving people to want to stay with the hierarchy as a path to the coveted spending power. But the real serious greed is normally found at the very top, and that is another discussion.

## Psychological Factors

### Challenge of Self-image

Daniel Kahneman writes about how we simply do not absorb facts that threaten our existing worldview or belief system. He writes about a company in the financial sector, where he showed that their highly elaborate scheme of bonuses were not based on skill but on pure luck.

> *The illusion of skill is not only an individual aberration; it is deeply ingrained in their culture. Facts that challenge such basic assumptions – and thereby threaten people's livelihood and self-esteem – are simply not absorbed. The mind does not digest them.*

A Danish politician at one time managed to say:

> *If these are the facts, I deny the facts!*

Often when we present Agile Lean Leadership, we get the question: "How do I implement this within the classic hierarchical line organization?". The fact that this stands in antithesis to the classic hierarchy has not sunk in. These people are neither stupid nor evil; it is just too much of a challenge to the current paradigm, so it is just dismissed. Kahneman continues:

> *I am convinced that the message that Thaler and I delivered to both the executives and the portfolio managers was instantly put away in a dark corner of memory where it would cause no damage.*

Our brain has the capability of just dismissing observations and fact that don't corroborate currently held views.

## Cognitive Biases

Daniel Kahneman is called "The king of human error". He lists countless cognitive biases that lead us in to error in judgement. Of interest to this discussion is in particular, these few:

1. Confirmation bias, which is the tendency to look for, favor, and find information in a way that confirms what you already believe.
2. Conservatism, which is the tendency to revise one's belief insufficiently when presented with new evidence.
3. Zero-risk bias, which is the preference for reducing a small risk to zero over a greater reduction in a larger risk.
4. WYSIATI is an abbreviation for "What you see is all there is". It is one of the human biases that Kahneman elaborates on. We believe we have seen it all, and we have built a compelling story around it. Dave Snowden talks about how we tend to view things as being in the "Obvious" domain, especially when we know little about them.

Kahneman concludes:

> *We know that people can maintain an unshakeable faith in any proposition, however absurd, when they are sustained by a community of like-minded believers.*

### Difficulty with Trust

The New Structure relies on trust. Not so in the hierarchy, it relies on control. More than once have I had someone above me say “Trust is good, but control is better!”.

If you haven’t trusted people for years then It seems too much to have to trust people, very risky. The classic hierarchy sets you up for a zero-sum game where if there are winners there must be losers, you compete, fight for “lebensraum”.

### Not Serious Syndrome

Often people react against the New Structure, simply because it appears less professional and serious than the hierarchical alternative. They experience the proposition to be too romantic and utopian, something from the seventies, like a surviving hippie subculture.

### The Comfort Zone

A lot of managerial positions are very comfortable. OK you have to put up with the occasional budget or abuse from above when a number falls out of range, but that passes and life settles down again.

You get used to it, and as Matthew Stewart says about the new aristocracy in the US, the top 9.9% of income spectrum, “It is good to be us!” (read more here…). As long as life is reasonably good, there is more to lose than to win. Why risk that with a New Structure? Why invite trouble inside?

## What can be Done about it?

In the end people choose what they want most in their heart, what really makes their life worth living. Although working in an organization hopefully is not all there is to life, it normally takes up about half of the time we are awake, so it does play a role.

For an Agile Lean Leadership transition to succeed, people who are onboard, have to want and desire this. Therefore, top management which holds the flag and the vision, must be very present and visible.

It must be shown to people – also those in the hierarchy – that a life of intrinsic motivation with focus on purpose, autonomy (freedom) and mastery for everybody is a better option than the Neo-Taylorist separation of thinking and doing which creates a new aristocracy.

It must be shown convincingly that working in teams and collaborating across Circles in a value stream can be very fulfilling also professionally.

It must be shown that adopting a servant leadership style, where those that make decisions of a broader scope than at the Circle or Team level, do it as a service to the primary circles, not as a privilege. Managers have to choose to become leaders, leadership is a service not a privilege.

The top management must change the compensation schemes and ways people are appreciated publicly, so that it becomes visible that the roles in the new structure are valued and appreciated. The system must be changed also in this area.

Everybody in the organization must commit to working towards transparency and trust, it takes time. Good old W. Edwards Deming said:

*It does not happen all at once. There is no instant pudding!*

For people to participate in a journey, they must see the attractiveness of the end state and also be willing to endure the travel. It is up to the leadership to create that attractiveness.

## Conclusion

If you create the New Structure – an Agile Lean Organization, there will be people who don't like it. It is an adoption of a different set of presuppositions. What if people don't want this? What if there is no desire or passion for it? Maybe people are too comfortable for change?

The people in the classic hierarchy, often referred to as middle managers, have several good reasons to avoid the change. The hierarchy has an inbuilt tendency to reward stability and predictability at the expense of innovation and experiment. This represents an extrinsic motivation of position, respect and compensation as you move up the steps, and as we know such extrinsic motivation overrides the intrinsic one. It is therefore perfectly understandable that these people prefer to stay in the hierarchy and not something to be scornful about.

It is up to top management to build the case for the New Structure as an attractive alternative; but it is also up to everybody else to be a bit humble about the change and demonstrate respect for people who exchange classic managerial positions with the roles in the New Structure and not lead them to feel demoted.

If you look at the people who set out to conquer the West in the US (forgetting for a moment the problem of the native Indians). It was not the people from back East who had the most comfortable lives, who set out on this journey, nor was it those who had significant positions to lose. These folks came later when there was something to exploit.

So, often it takes a stimuli from the outside to ignite a transformation; a challenge that just cannot be ignored. We have in other articles mentioned General Stanley McChrystal and Commander David Marquet; for both of them the trigger was a realization of the fact that the status quo was not tenable any longer.

People who have much administrative work will typically experience a significant change in their daily lives as there simply is much less of that kind of work in an Agile Lean Leadership organization. Very often I have found that people really like to be more focused on skills that contributes to the value stream. Administration is waste, some of it necessary waste, but usually a lot of it just serves as fuel for the hierarchy.

There is one class of people though who are deers in the headlight in such a transformation. Those whose only skills are bureaucratic. That is, they only know to relay orders down and reports up. They seriously need to be retrained or old skills need to be resurrected. There is precious little use for the bureaucratic skill in an Agile Lean Leadership organization.

At the end the proposition is to create an organization that is effective, fast, sustainable and resilient. It is there for those who want it.

***

To read this article online with embedded hypertext links, go here:

**https://www.linkedin.com/pulse/attraction-classic-hierarchies-kurt-nielsen/**

# About Kurt B. Nielsen

Chief Executive Officer and Director

**Kurt** B. Nielsen is CEO and one of the founders of AgileLeanHouse A/S (ALH). ALH has a vision of helping people in organizations get more out of their efforts. ALH does this by teaching Agile and Lean disciplines, coaching and support organizations on site and developing tools, among these the flagship product Agemga (https://agemba.com).

Kurt has been a Certified Scrum Trainer under Scrum Alliance for more than 10 years and taught thousands of Scrum Masters and Product Owners.As a prolific writer he has produced a number of very challenging and inspiring articles, see LinkedIn, Medium and the ALH website.

Kurt is known as an effective and inspiring presenter, teacher and coach, people frequently report huge impact from their interactions. Kurt combine commercial, technical, human resource and ethical views on leadership.

For more information, visit https://agileleanhouse.com/ or connect at LinkedIn: https://www.linkedin.com/in/kurtbnielsen

# IN PRAISE OF SWARMING

By Dan North

Posted on January 26, 2018

Most of my work these days is helping organisations figure out how to be more effective, in terms of how quickly they can identify and respond to the needs of their external and internal customers, and how well their response meets those needs. This tends to be easy enough in the small; the challenges appear as we try to scale these techniques to the hundreds, thousands or tens of thousands of people.

It is into this space that a new generation of software methods have emerged. SAFe, LeSS, DAD and others claim to help enterprises "scale agile," whatever that means. A generous interpretation is that people who have a track record helping organisations on this journey have managed to codify their knowledge into a set of blueprints, guidelines, frameworks and methods so you don't have to. Another take is that execs in organisations above a certain size like to buy a packaged solution to a perceived problem, and people with experience in selling to this market know exactly how to dress something up to look sufficiently appealing.

The most successful transformations I have experienced—the ones where you walk into the office and there is a tangible difference in the energy and interactions between people, where the commercial and management stakeholders are as excited and invested as the technology stakeholders, where everyone agrees on the metrics that matter and those metrics are trending in the right direction, where the people involved are proud to be part of the change, and where those outside are eager to get involved—those transformations succeed because a small cabal of dedicated individuals take the time and effort bringing people on the journey with them; they create and nurture a groundswell of enthusiasm; they engage, challenge, educate and support the middle and senior managers whose world is being turned on its head; they gain the executive sponsorship and investment to carry the whole thing forwards; they use external help because they know they can't do it all themselves. And they do all this without recourse to canned methods or certification Ponzi schemes.

I think this approach deserves calling out so I am offering a new acronym, *SWARMing: Scaling Without A Religious Methodology*. My argument isn't that packaged scaling methods are unhelpful per se, rather that they are *neither necessary nor sufficient* for successful transformation. They can be anything from a useful starting point to an expensive distraction, but one thing they are not is a solution.

## No silver bullet

The hard truth is that most lean or agile transformation initiatives will fail, because they challenge the very fundamentals of the organisation: how the money works, how governance works, how management works, how accountability works. These transformation programmes tend to happen in 3 to 5 year cycles. With each change cycle a senior manager adopts the latest iteration of words and rituals without taking on the hard work of engaging with the organisation's underlying structures, behaviours, values and beliefs, and three to five years later the latent constructs have prevailed. The only tangible results are a large transfer of cash to the external consultants, a rash of certificates with a recurring revenue model, and a metaphorical shrug of the corporate shoulders. The inevitable hand-wringing that ensues and the need to do something precipitates the next change cycle, and the wheels turn once again.

Transformation is not a transactional activity that starts and ends and has a budget, and it should not be treated as one. It is never finished, in the same way evolution or adaptation is never finished. The goal is rather to break free from the thinking that led to the incumbent system by introducing a new paradigm, and then to use this to achieve a new steady state where the organisation is responding quickly and effectively to external and internal feedback, behaving like a genuine learning organisation. The challenge is to encourage, precipitate, and then support that paradigm shift. The principles underlying the new model are straightforward to describe, but they involve a root-and-branch rethink of the fundamental structures of the organisation.

I see a number of factors necessary for genuine, measurable, lasting transformation. Where I've seen transformation attempts fail, one or more of these has been conspicuously absent. Before I discuss these I want to explore some of the differences between the old world and the new, and look at what makes these packaged scaling frameworks so appealing.

## The shape of waterfall

A paradigm is the set of assumptions or beliefs we hold to be true. Over time we institutionalise these into corporate policy, organisational structures and even into law, as I have discussed elsewhere.

I want to highlight some of the differences between the old and new paradigms to show the ground we have to cover. In each section I first describe the traditional paradigm and then the corresponding assumptions in the new world. As you read each paragraph, whether or not you agree with the statements, remind yourself that these organisations are acting as though they are true.

### Cost efficiency: utilisation vs flow

In a traditional firm the organisation is seen as a number of separate departments, each of which is either a cost centre, like IT or HR, or a profit centre like Sales. The difference is that a cost centre doesn't contribute directly to revenue. Costs are allocated to each cost centre, and cost efficiencies come from reducing the operating costs in the cost centres and maximising the revenue from the profit centres. Common strategies for reducing costs, especially towards the end of the financial year, are limiting travel, reducing variable headcount by laying off contractors and other temporary staff, shifting work to cheaper countries or locations ("moving the work to the people"), repurposing legacy hardware and equipment, and forcing teams to use standard components to minimise "cost per use". Common strategies for maximising revenue are offering commissions on sales, setting aggressive quarterly targets, and offering incentives and rewards for achieving specific goals. Gains come from maximising resource utilisation, i.e. keeping people busy.

In an agile organisation—in the broader sense of business agility—the organisation is seen as an interconnected system where all departments are value-generating, either directly contributing to the final product or indirectly enabling the organisation to work more effectively. Value chains map the creation of value through the organisation, ending with meeting a specific customer need. Costs and revenues are allocated holistically across the whole value chain, and cost efficiencies come from optimising the flow of value. Common strategies for reducing costs are identifying and eliminating non-value adding activities, limiting the amount of work in process to reduce hidden inventory of work, and assembling multi-disciplinary teams to reduce handoffs and other delays ("moving the people to the work"). Common strategies for maximising revenue are working iteratively and incrementally in small batches to realise value sooner and improve Risk-adjusted Return on Capital, frequent testing of product ideas with customers to identify which work not to do, and enabling teams to choose the most effective tools to reduce their time to market. Gains come from optimising flow efficiency, i.e. keeping work moving.

### Managing work: project vs product management

In a traditional firm scope is defined in detail and locked down before the work starts, usually as the result of a project or programme initiation exercise, in the form of a Business Requirements Document as part of a Business Case, which will also contain a cost and resource breakdown and revenue projections. This is presented to a funding board and the programme budget is allocated and signed off. Success is defined as delivering the agreed scope within the cost and time constraints. Variance in scope is frowned upon, as is replanning, which can have severe knock-on effects elsewhere in the organisation.

In an agile organisation scope is defined dynamically within a product area, usually in terms of moving a business metric in a desired direction, for instance reducing customer churn or increasing profit per customer visit. Funding is allocated as an ongoing run rate for the product area, and revenue is tracked as a near-term trailing indicator. Success is defined as the ongoing achievement of the business metric goals, or learning through evidence that this is the wrong strategy. Variance in scope is both expected and encouraged, and regular replanning based on market and operational feedback is normal.

### Portfolio planning and governance: up-front vs ongoing

In a traditional firm the annual plan is sacrosanct. People up and down the organisation invest considerable time and effort in the annual budgeting and planning cycle. Each department predicts what it will spend and what it hopes to achieve, and a central body determines each department's budget. These budgets cascade into the programmes and projects within each department and a centralised PMO or similar function monitors and polices progress. Replanning has a high cost and effort and is discouraged, so there are repercussions for deviating from the plan once it is underway. Projects and programmes are assessed as Red, Amber or Green ("RAG" status) according to their level of deviation from plan, and the project and programme managers are held to account accordingly.

In an agile organisation financial planning happens more frequently, typically using a rolling one year plan reviewed quarterly. Many firms are adopting OKRs as a mechanism for this. The PMO supports global system optimisation: Green means things are good; Amber means a project or programme needs help with a decision that has implications beyond its boundary; Red means the lack of such a decision is putting the outcome materially at risk. "RAG" status in this context is about how well the project or programme is being supported by the wider organisation.

### The appeal of frameworks

There are other contrasts I could draw but the above should serve to illustrate the cultural and organisational chasm we face and highlight the work ahead of us. Sociologist Ron Westrum goes into more detail about different organisational cultures but these are some of the more common differences I have encountered.

Packaged frameworks are appealing because you can easily find this overwhelming. You want this Business Agility magic you keep hearing about but you don't want to, or are unable to, challenge the corporate funding and governance processes, and whatever you do has to fit within strict departmental cost controls. So you start by sending a handful of project managers for Scrum Master certification, and maybe some business analysts for Product Owner training, then you try Scrum for a bit. Later you might try Scrum-of-Scrums, SAFe, LeSS, or similar with the best of intentions. If you can just to prove it works in a contained pilot environment, the rest of the organisation will surely wake up and beat a path to your door!

The problem is that this kind of transformation requires a fundamental shift in how people think and how they operate. This in turn requires acknowledging that the way they are running things is based on assumptions and constraints that are no longer relevant to the modern business environment; that they are still in business in spite of, rather than because of, these inherited management and governance systems. No one likes to acknowledge that, even to themselves. Their subconscious minds vehemently resist this paradigm shift, which makes them easy prey for a vendor selling a product that promises said transformation while conveniently fitting their existing paradigm.

The vendors and consultants can do this regardless of whether it will deliver the results, safe in the knowledge they and their methods won't be held accountable if it doesn't, because the one thing everyone can agree on is that *transformation is hard* and at least we tried. Many proponents of these methods have a religiosity about them. Their method works; if you don't believe this you are misguided, misinformed, or just antagonistic (I've been accused of all three); and if it doesn't work then you are applying it incorrectly.

## Scaling Without A Religious Methodology

There are a handful of ingredients without which you are unlikely to achieve lasting change. They are by no means a formula for success. Rather, you should consider the absence of one or more of these a danger sign, a significant risk to be managed and mitigated.

### 1. Time

Transformation takes time, patience and tenacity. Shifting a large enterprise of thousands or tens of thousands of people will take a number of years. The technology adoption lifecycle applies here, with its pioneers, early adopters, early and late majority, and laggards. You should focus your efforts on the pioneers and let their successes draw in the early adopters, and so on. Too many transformation efforts try to reach everyone at once—"We are going agile from Monday!”—and the social demographics show that this just doesn’t work. You should expect no more than 5% of your organisation to be pioneers of this kind of change, so work on identifying and recruiting them, and giving them the environment and tools they need to succeed.

## 2. Investment

Just as this transformation won’t happen overnight, it also won’t happen on its own. You will need to assemble a dedicated team, which means ring-fencing and funding them, and delegating or managing their responsibilities. In the greater scheme of things this is a tiny investment—you can get a long way with a handful of people—and I am continually surprised at how difficult it can be to get this allocation of people and funding in an organisation that clearly wants to go on this journey. It is worth taking the time to ensure this commitment is in place before you get going, however long it takes.

## 3. Communications

Different audiences respond to different messages. Take the time to understand your stakeholders and think about the kinds of communication they will find useful: Which channels? How often? What content? What is the goal of each communication? How can you meet them where they are? Where do you want them to be? Where are the gaps? From awareness roadshows to executive briefings, the way you structure and communicate your vision, aspirations and progress will be most people’s primary view into your world.

At the very least you should be able to answer “What’s in it for me?” and “What do you need from me?” for every stakeholder group. As well as the people you will be directly interacting with you should also consider anyone indirectly affected by the transformation. These could be internal and external suppliers, customers, partners, regulators, auditors, or anyone else

You will have your share of detractors, primarily the late majority and laggards who have a vested interest in maintaining the status quo, and who will be anything from indifferent to critical to actively sabotaging your efforts. And don’t be surprised to find allies in unexpected places. Amplifying your successes and neutralising your opposition can take up a lot of your time so having a robust communication plan in place will help you stay on track and not lose sight of important stakeholders.

## 4. Influence

More than anything else your core team needs both the ability and the credibility to influence up, down and across the organisation. Trying to do things on your own will be like boiling the proverbial ocean. You are trying to create a groundswell and encourage change, not do it all yourself. You need the requisite executive sponsorship, and people who are not afraid to speak truth to power, respectfully and professionally, and who know when to shut up and listen.

Some years ago Lean Operations veteran Richard Durnall wrote an article describing the arc of transformation an organisation goes through. That link is a slide from a talk I gave that includes a discussion of this arc, because Richard’s blog is no longer online. He describes six stages of transformation in terms of things that break. You start by introducing new ideas that challenge incumbent beliefs (“The people break”). Soon you find the processes and tools are no longer fit for purpose (“The tools break”). Measuring flow of value rather than busyness and effort brings you into conflict with the PMO (“The governance breaks”). As you start to scale you realise just how many stakeholders you need to bring on the journey with you (“The customer breaks”). Shifting to product rather than project management means you need regular incremental funding rather than infrequent, big investments, which the corporate finance model isn’t set up for (“The money breaks”). Finally you realise that silos of functional departments are exactly the wrong structure for business agility (“The organisation breaks”).

The two most challenging stages are around the governance and the money. This is where corporate structures, policies and culture are likely to be the most ingrained, and where you will potentially encounter powerful friends and adversaries! The sooner you can start to influence the governance and finance elements, the greater your chances of lasting success. Conversely if one or both of these are outside of your control, your potential sphere of impact will be correspondingly smaller.

## 5. Education

There are some core fundamentals that you and your pioneers will need to learn, practise and internalise. My starter list would include understanding the basics of Theory of Constraints, Systems Thinking, Throughput Accounting, Complexity Theory, Lean Product Development, Continuous Delivery, modern leadership, and change. If you are a book learner you can glean most of this from a handful of well-written books:

- *The Goal by Eliyahu Goldratt*
- *Thinking in Systems: a Primer by Donella Meadows*
- *Lean Enterprise by Jez Humble, Barry O'Reilly and Jo Molesky*
- *Continuous Delivery by Dave Farley and Jez Humble (again!)*
- *Turn the Ship Around by David Marquet*
- *Fearless Change by Linda Rising and Mary Lynn Manns, and for extra credit*
- *The Principles of Product Development Flow by Don Reinertsen.*

The key here is understanding the core principles without getting bogged down with specific implementations. This is where those scaling frameworks can be helpful, as a "non-blank piece of paper" to demonstrate to your sponsors that you aren't just making this up, and to provide an initial vocabulary. But take the time to understand the principles behind the methods and techniques so you can adapt them to your own context.

## 6. Practise

"Knowledge without practice is useless. Practice without knowledge is dangerous," says Confucius. Reading about these topics is like reading about scuba diving or surgery. You won't appreciate it until you do it for real, and like any new set of skills it will take time to get good at it. Messing up is a necessary part of learning. You should therefore think about how to practise in a safe-to-fail way. That conversation you want to have with the CFO about releasing smaller amounts of funding more frequently with less of the heavyweight approvals boards and governance theatre? You should probably practise that first on a friendly finance stakeholder who will politely let you know they glazed over as soon as you mentioned velocity and story points. And that they don't think the CFO will appreciate the phrase "governance theatre."

In the words of a famous Swedish poem, you should *speak with farmers in farmers' words, and with learned men in Latin*, in other words practise adjusting your message to your audience. Likewise try applying flow tools like value stream maps, lead time measurement and cumulative flow diagrams on a small, friendly project first, before you try to roll it out across the wider programme. Gather evidence before you pitch flow metrics to the PMO. You may only get one chance.

## 7. External help

My favourite definition of a good consultant is someone who has screwed up enough times before on someone else's dime that they don't have to do it on yours. You can go it alone but you are likely to fall into all the traps everyone falls into, and your fundamental attribution error bias means you won't believe me. There are two kinds of external help that seem to make a real difference: trusted advisers and embedded coaches. The former will be invaluable early on and should become less necessary over time. The latter you want to engage as you start to build momentum and realise you can't be everywhere at once. The problem is that anyone can declare themselves an agile or lean coach just by updating their LinkedIn profile—around 18,000 at time of writing—so it is worth shopping around and doing the due diligence. Use your network—word-of-mouth recommendation and track record are by far the best selection criteria.

As you start to build momentum you don't want an army of self-declared coaches with different levels of ability and experience, different approaches, ideologies and allegiances fragmenting your efforts. I have seen more than one transformation initiative hoover up all the agile coaches they could find, only to eject them all some time later with little progress having been made.

Frustratingly this is one area where certification could help, by assuring a level of competence, consistency and experience, but that would require peer-appraised, practise-based certification and assessment, and there is no network of peers that could guarantee any degree of quality or consistency here, especially between the various competing scaling methods.

### 8. Consistent, invested, resilient leadership

From the top of the organisation down, the leadership culture will make or break an agile transformation. There are many adjectives that could describe good leadership. I have chosen these three as my key ingredients, although I'm sure others will have different and equally valid opinions.

**Consistent leadership** is in it for the long haul. Various sources tend to agree the average tenure of a CIO is around 4-5 years, which means a good proportion of them will be moving after 2-3 years. (There is a tendency for CIOs approaching retirement age to stick it out and skew the graph to the right.) I have seen otherwise successful transformations stalled and then abandoned after a change of senior leadership, especially when someone is brought in to "control costs". They invariably interpret this as slashing departmental budgets rather than trying to figure out why so much effort produces so little value. Ironically they would gain much greater savings by following through on the transformation.

I had the pleasure of meeting Henk Kolk, former chief architect at ING Bank, in 2015 when he was talking about their agile transformation journey. When we met this had been going for some years and was still well underway. A recent McKinsey article shows that although some of the names have changed, the new incumbents have carried forward the same culture and commitment.

**Invested leadership** realises this is the future of the organisation, not a side project. In Theory of Constraints terms, the management team chooses to define business agility as the goal of the organisation and makes everything else subordinate to this. Senior execs can get distracted, often by external market or political factors beyond their control, so it is important they have a clear and vocal commitment. At one bank I was working with the CEO ran a kanban board with his management team. They used this to make work visible and to limit the amount of work in process and it was the focus of their daily standup. This said more about his commitment to business agility than any Town Hall address.

**Resilient leadership** is bigger than a charismatic chief exec. For leadership to be resilient you need leaders up and down the organisation, all committed to a common goal. I advocate two complementary leadership styles, namely *servant-leader* and *leader-leader*.

*Servant-leader* describes a dual role. You are both a servant to your team, ensuring autonomy, and anticipating and meeting their needs, and their leader, setting direction and holding them accountable. Many people professing servant-leadership forget this latter part and just see their role as "protecting" the team, but without direction you lose the alignment across teams. As Kent Beck said, "autonomy without accountability is just vacation."

*Leader-leader* is a term coined by former US navy commander David Marquet in his excellent *Turn the Ship Around*, one of my favourite books on leadership. Each person in the organisation considers themselves the leader of their own area and "expresses intent rather than asking for permission." In Marquet's words, the idea is to "move authority to the information" by delegating autonomy, rather than "moving information to the authority" and having decisions bottlenecked at a poorly-informed, senior decision-maker.

## Concluding thoughts

If you are already SWARMing you have my respect and admiration. I hope this article helps you identify gaps in your strategy, or at least reaffirms that you have the bases covered. As with any complex adaptive system, the factors I have outlined are all interrelated. Influence can buy you time and investment. Investment will raise the profile of your efforts and can give you more influence. Resilient leadership creates an environment that encourages practise, and so on.

If you are considering a journey towards business agility, be aware there are alternatives to wholesale adoption of scaling frameworks, with their recurring revenue certifications, closed community of self-professed experts, and methodologies that simultaneously promote inspecting and adapting whilst insisting on strict adherence to their branded processes, tools and methods. At best they can cover only a fraction of the landscape I describe above. The rest is down to you and the people you choose to travel with.

Methods like SAFe, LeSS, DAD and others are based on sound principles and may make a useful starting point, for example I have seen DAD used in a number of places to create a consistent initial vocabulary and terminology. But the best person to change your organisation is you, and the best way is however you figure.

***

To read this article online with embedded hypertext links, go here:

**https://dannorth.net/2018/01/26/in-praise-of-swarming/**

# About Daniel Terhorst-North

**Daniel** Terhorst-North uses his deep technical and organisational knowledge to help leaders, managers and software teams to deliver business results quickly and successfully. He puts people first and finds simple, pragmatic solutions to organisational and technical problems, often using lean and agile techniques. With thirty years of experience in IT, Daniel is a frequent speaker at technology and business conferences worldwide. The originator of Behaviour-Driven Development (BDD) and Deliberate Discovery, Daniel has published feature articles in numerous software and business publications, and contributed to *The RSpec Book: Behaviour Driven Development with RSpec, Cucumber, and Friends and 97 Things Every Programmer Should Know: Collective Wisdom from the Experts*.

He occasionally blogs at https://dannorth.net/blog.

# On Scrum Mastering...9 things to try

By Ewan O’Leary

Posted on September 26, 2018

Many who find themselves thrust into the role of Scrum Master struggle with how to show up and guide their teams towards self-organization. Here are a few suggestions that might give you the support you need in making the change. Remember these are offered as things to try, experiments if you will, rather than prescriptions, so hold them lightly! You may find that your team is significantly less receptive to them than you are. When you introduce these experiments, try to include your team by making them aware of what you’re doing. This helps to break the ice and gives them confidence to try experiments of their own. With that said, here they are:

1. **"I’m Getting Coffee":** When you’re in the midst of running your daily standup, you should feel as if you could leave at any moment and get coffee. Try it! Let your team know what you expect, and then leave the room, allowing them to engage with each other without thinking they’re performing for you, or you feeling like you have to do something. This should feel a little uncomfortable and that is ok! This is you learning a different way of being in service of your team - remember, you need to allow the flow of their conversation and they need to own the meeting outcomes. **Bonus:** Offer to get them coffee or refreshments.
2. **Task Focus vs Value Focus:** Is your team focused on tasks? Are you walking the board based on individuals, rather than stories? Remember, your team isn’t there to do work, although doing work is helpful! Rather, they’re there to deliver value. Orient all your conversations around delivering value ie completing the user stories, as opposed to completing tasks ie doing work. See what happens to your ability to complete stories when your team is focused on goals!
3. **Stop Talking and Start Helping:** Sometimes the best thing you can do is to stop talking. It’s often the most helpful move to sit in an awkward, uncomfortable silence while the team figures out what to do. Ask an open-ended question and let them speak first, no matter how long it takes! Don’t be afraid to get uncomfortable. In fact, your effectiveness as a coach is directly related to your ability to be comfortable with others discomfort, and help them see that their discomfort is really their growth edge, that place where they’re building important new muscles that will help do things better.
4. **Don’t Think For The Team:** The more you think for your team, the less they think for themselves. Remember this - your job is NOT to solve their problems. It is to create the capability in the team to solve their own problems. This is a very, very important distinction - you don’t ever want to be thinking for them, especially when they are not used to taking up responsibility for themselves (like in many organizations with a strong command culture, where people comply with instructions and thinking differently is frowned upon:). If you have a project management background, this can be one of the most challenging things to let go of, so give yourself a little slack around this, and recognize that you probably trained yourself to do exactly the opposite and biased yourself towards action on behalf of the team when you saw things that might be going wrong.
5. **Do Less:** The less you do, visibly and intentionally, the more the team will need to do. Often, new scrum masters step into the role full of enthusiasm and energy, and end up becoming the team’s administrative assistant. Rather than facilitating the process, there is a tendency to carry the team’s energy. Unfortunately, this often has the opposite effect, disengaging your often remote team members rather than inviting them in to a participative,

collaborative relationship. So let them know that you’re going to be doing less, especially in the standup, and see how it feels - remember you have many standups (up to 10) per sprint, so there is a lot of room for trying different approaches.

6. **Reduce Specialization by Increasing Learning:** If your team is struggling to complete user stories because you have dependencies on one person either in your team or another team, then its your responsibility to raise or resolve the issue. One pattern I suggest is for you to create a list of skills needed to complete tasks when you become aware that they are in short supply, and then use this as input for team learning.
7. **Hub and Spoke vs Mesh:** Are you the center of the conversation, or just supporting it from the side? The shortest distance between two points is direct - so draw the flow of conversation on a Flipchart or online whiteboard. Draw a circle at the center - thats you. Draw a circle for each person on the team around you, like the time on an analog clock. Every time someone speaks to someone else directly, draw a line. The best meetings are those where there are the most lines between team members, and the fewest towards you. This will visually show the team the flow of communication in your standup! Very useful to make this visible. You’re looking for a mesh of conversations between team members, not you at the center of the hub and spokes.
8. **The More You Do The Less They Do:** If you’re going to do the heavy thinking around coordination and risk mitigation (like you might have when you were a project manager working with SMEs on specific tasks individually) the team isn’t going to learn that that is really their job. Rather than thinking for them, think about ways to frame their thinking. Help them with time boxing their discussions, even posing the challenge - how might we solve this problem if we only had 5 minutes to do it? Or something similar.
9. **The Team Is Smarter Than You:** Collectively, the team is going to be smarter than you. Your job is to harness their minds to a collective purpose, and to do that, you must allow them to be autonomous, and create a team culture in which they will be inspired to develop mastery of their craft.

Remember, be curious, be open - this is a journey! Be compassionate with yourself, and with others, as you experiment with these suggestions.

***

To read this article online with embedded hypertext links, go here:

**https://www.linkedin.com/pulse/scrum-mastering9-things-try-ewan-o-leary/**

# About Ewan O’Leary

**Ewan** O’Leary is an organizational and leadership coach, facilitator and trainer with Future State, based in San Francisco, CA. He has worked with a diverse group of clients including Paypal, Nike, Genentech, Thomson Reuters, Rockwell Automation, Express Scripts and Stubhub. Ewan has more than 20 years of experience supporting people and organizations around the world in finding new ways of working and engaging with their stakeholders.

Through TealShift, Ewan is the creator of TealWoWSM, an integral approach to work incorporating elements of Agile, design thinking, lean startup and developmental coaching to allow organizations and individuals in them to liberate their potential.

An active participant in Agile community events, Ewan holds IC-Agile Expert certification, Scrum Alliance certifications and is a regular contributor to the ongoing development of Integral Agile. He blogs occasionally at www.ewanoleary.com and at writes at www.tealshift.com.

# 5 Powerful Things About the Sprint

By Stephanie Ockerman

Posted on January 21, 2018

The Sprint is one of the five Scrum events.  In my Professional Scrum Courses, this is the event that people often forget about because it is a container event, not necessarily something you distinctly schedule on the calendar.

This container holds the space for all of the work to create the shippable Increment of product, and it is limited to one month or less in duration (i.e. time-box).  This container starts with Sprint Planning and ends with the Sprint Retrospective.  Then the next Sprint immediately starts.

The Sprint can seem like a simple administrative term, and people sometimes brush it off.

**However, the Sprint holds so much power in Scrum.**

This is why the Scrum Guide calls the Sprint the "heart of Scrum."  Let's take a look at 5 powerful things the Sprint provides teams and organizations.

## #1 – Focus

The purpose of a Sprint is to create a **potentially releasable product Increment of value** to the organization.  It's that simple.

What value is to be delivered is guided by the Sprint Goal, which does not change during the Sprint.  Because… well, focus.

If you've worked in product development for even a day, you probably understand how chaotic it can feel.  New ideas and new business needs are popping up.  New information about the market and customers is being discovered.  And of course, the complexity of the actual work a team is doing creates a continuous flow of new learning and new challenges.

By having this single purpose every Sprint – to create a releasable Increment – the team can bring their focus back to this.  They can **set aside the distractions not related to this purpose**.  They can take the new information they have uncovered and adapt their plan without losing focus.

This brings us to predictability.

## #2 – Predictability

While a Scrum Team may not be able to guarantee the specific scope of the Increment (i.e features/ functions), **a team that is using Scrum well will be predictable in delivering a "Done" Increment** every Sprint.

Sprints have a consistent cadence.  This **consistent cadence** helps a Scrum Team understand what they are capable of delivering in a period of time.  As a Scrum Team understands this and continues to work together, they can better forecast delivery expectations.

Disclaimer:  Remember that the words "estimate" and "forecast" imply there is still complexity and uncertainty, and these estimates and forecasts will not be perfectly accurate.  The Sprint helps us **optimize predictability over time**.

A Scrum Team can change the length of a Sprint, but they don't change it constantly.  They do so intentionally as a part of their commitment to continuous improvement in meeting the business needs.  Then they learn and settle into a new cadence.

And that brings us to #3.  The Sprint provides control.

## #3 – Control

A question I am often asked is,

*How long should our Sprint be?"*

My answer is always,

*How frequently does your business need to change direction?"*

Remember the Sprint Goal does not change during the Sprint.  This provides the stability a Development Team needs in order to get something meaningful done (see #1 above).  So the real driver of the length of a Sprint is **how often the business needs to inspect the Increment and adapt the direction based on new information**.

This gives the business control without creating an unstable situation for the Development Team.

In addition, the **Sprint time-box gives the business more transparency to and control of cost and schedule.**  An organization can fund a number of Sprints and see the value they are getting every Sprint.  This helps make informed decisions about whether or not to keep investing money and on what to invest it.  Ultimately, this is how you control risk in complex environments.

## #4 – Freedom

A Sprint provides freedom.  This may seem contradictory to point #3, but it is not.  And that is the beauty of a Sprint.

The Scrum Team has the focus of a Sprint Goal and a time-box.  These **boundaries create the freedom for effective self-organization, collaboration, and experimentation**.

There are many ways risk shows up in product development.  Are you building the right thing?  Are you building the thing right?  What assumptions might be wrong?  What might change?

## Teams have to learn by doing, inspecting and adapting along the way.

And businesses do too.  The business gets the freedom to experiment as well.

Sometimes there will be failure.  In fact, failure is a part of learning.  The question is how big of an impact that failure will have. **A Sprint limits the impact of failure to the time-box of the Sprint**.

This freedom leads us to opportunity.

## #5 – Opportunity

Scrum is a framework for opportunistic discovery.  To quote Ken Schwaber, it helps us “harness change for competitive advantage.”  Ultimately, successful Sprints enable the benefits of business agility.

Scrum is the art of the possible.  It’s about being open to and ready for the opportunities you discover throughout the journey.

***

To read this article online with embedded hypertext links, go here:

**https://www.agilesocks.com/5-powerful-things-sprint-scrum/**

# The Art of Product Backlog Refinement

By Stephanie Ockerman

Posted on May 20, 2018

A common question I hear in Scrum training courses and in coaching sessions is, “how much Product Backlog refinement should we do and how much detail should be in the Product Backlog?”

First, let’s look at the Scrum Guide.

## Product Backlog Refinement

According to the Scrum Guide, Product Backlog refinement is the act of adding detail, estimates, and order to items in the Product Backlog.  But Scrum doesn’t prescribe how you do it, and for good reason.

Refinement is an ongoing process.  It never stops because requirements and opportunities never stop changing.  Detailing everything up front would create waste and also delay the delivery of value.

## Different products and different teams will have unique needs in terms of frequency, techniques, and level of detail.

Even the same Scrum Team working on the same product will need to evolve how they do Product Backlog refinement over time to fit new situations.  Scrum Teams needs to figure out what works for them.  So how do they do that?

## Self-organization and empiricism.

Apply the *Goldilocks Principle* to help a team experiment and find what works best for them through inspection and adaptation.

## The Goldilocks Principle and Product Backlog Refinement

The *Goldilocks Principle* is about finding what is “just right” for you.

## The goal is to balance gaining enough benefits from the activity while

**minimizing the potential waste.**

*Let's first look at the 6 benefits* of Product Backlog refinement:

1. Increase transparency
2. Clarify value
3. Break things into consumable pieces
4. Reduce dependencies
5. Forecasting
6. Incorporate learning

Now let's dive deeper into each of these to see how we can apply the Goldilocks Principle. I'm going to give you a couple of starter questions in each of these 6 benefit areas to help your team figure out if it's too hot, too cold, or just right.

### #1 – Increase Transparency

The Product Backlog is an artifact that helps provide transparency. It is the "single source of truth" for what is planned in the product. Adding details increases transparency to what you plan to deliver, as well as your progress.

*Goldilocks Questions*

- How well do stakeholders and the Scrum Team understand what is planned for the product?
- How frequently are the interested stakeholders surprised by what was delivered?

### #2 – Clarify Value

When you clarify the details around value, the outcomes you are trying to achieve with the Product Backlog Item (PBI) are more clear. *Why do you want this? What is the user benefit? What is the business benefit?*

This helps **the Development Team build the right thing** to meet the need. This can affect what is requested, the estimate, and the order as the Product Owner and Development Team figure out what is actually needed. This conversation creates a shared understanding.

*Goldilocks Questions*

- How often do you discover during a Sprint that there is not a shared understanding of the business need or what you are building to meet it?
- How frequently do you discover in a Sprint Review or after a release that a PBI does not meet the user or business need?

### #3 – Break Things into Consumable Pieces

You want PBIs to be small enough that a Development Team can complete more than one in a Sprint. Having more than one PBI in a Sprint gives the team some flexibility to meet a Sprint Goal and deliver a "Done" Increment.

*Goldilocks Question*s

- How often are you not delivering a "Done" Increment? How often are you not meeting a Sprint Goal?
- When is this attributed to discovering mid-Sprint that PBIs are much bigger than you thought or not sliced thin enough?

### #4 – Reduce Dependencies

Dependencies often turn into impediments and can grind a team to a halt. While you may not avoid them all, you should try to reduce them where possible. This is especially important for dependencies outside the Scrum Team.

You can slice and split the PBIs in different ways.  You can re-order, or you can collaborate with other teams to help resolve the dependency in advance.  There are many options, and at the very least, you want the dependencies to be transparent.

*Goldilocks Questions*

- How often do you discover dependencies during a Sprint that jeopardize the Sprint Goal?
- How long do PBIs in a Sprint stay “blocked” by dependencies?
- When do you have to re-order the Product Backlog to account for dependencies?  And how much of an impact does this have on the Product Owner’s ability to optimize value?

### #5 – Forecasting

A refined Product Backlog combined with historical information about the Scrum Team’s ability to deliver working product helps you forecast.  Some products need to forecast several Sprints into the future to help communicate release expectations with stakeholders.  Other products will not have a need to do forecasting beyond the current Sprint.  Most products fall somewhere along this spectrum.

Related to forecasting, you also may need a refined Product Backlog in order to get funding.  **Scrum does not forbid up-front planning.**  Scrum simply says to consider your effort to do so, the potential waste, and the fact that you cannot perfectly predict the future in a complex domain no matter how much analysis you do.

*Goldilocks Questions*

- How much lead time is necessary for users, customers, and other stakeholders to implement a new feature or function?  What is the impact if they have less lead time?
- How much detail do users, customers, and other stakeholders need in release forecasts?  What is the impact if they have less detail?

### #6 – Incorporate Learning

Empiricism is about incorporating the learning you gain as you build the product, as you better understand how to realize the product vision, as you see changes happening in your environment.

*Goldilocks Questions*

- How are you adapting the Product Backlog to reflect new learning about the product’s evolving capabilities and how users are responding to the changes?
- What opportunities have been missed?  What prevented you from responding sooner?

## Pulling It All Together

You’ve discussed the Goldilocks questions about refinement benefits with the Scrum Team.  (Sprint Retrospectives are a great opportunity to regularly have these conversations.)  Now it’s time for the Scrum Team to decide how to adapt their process for Product Backlog refinement.  There is a reason these are open questions and not simple yes/no questions.

You are looking for balance, or that “just right” spot.  You want to achieve enough of the benefits of refinement while minimizing your waste.

With the information gained from exploring 1-6, the Scrum Team can now consider these questions **with the balance of benefits and waste in mind.**

*Goldilocks Questions*

- How frequently do you want to do refinement?  And how much time do you want to spend detailing the Product Backlog?

- Who do you want to be involved in refinement?  What knowledge and perspectives are needed?  How will you enable shared understanding?
- How much of your Product Backlog do you want to be “Ready” before a Sprint?  What does “Ready” mean to you?
- How do you want to communicate important details about PBIs?  What methods are working well and what methods are not?
- How will you ensure you can see the whole and not get bogged down in details?

***

To read this article online with embedded hypertext links, go here:

**https://www.agilesocks.com/product-backlog-refinement/**

# About Stephanie Ockerman

**Stephanie** is a Scrum.org certified Professional Scrum Trainer (PST), Scrum Master, and Co-Active Coach. She is a Curriculum Steward for Scrum.org's Professional Scrum Master (PSM) Program, working with the international trainer community and Scrum co-creator Ken Schwaber to carry forward the vision. She has over ten years experience delivering valuable products and services and facilitating training for professionals.

Stephanie is a results-driven, compassionate person who enjoys working with people who believe in the art of the possible. She started her agile training and coaching business Agile Socks LLC to fulfill this purpose. Stephanie combines training and coaching to help people level-up their skills and amplify their impact. She feels lucky to combine her passion for teaching, servant leadership, and experiencing the world by teaching and speaking internationally. You can read Stephanie's musings on Scrum, agility, and life on her blog at AgileSocks.com.

# Eliminate dependencies, don’t manage them

By Illia Pavlichenko

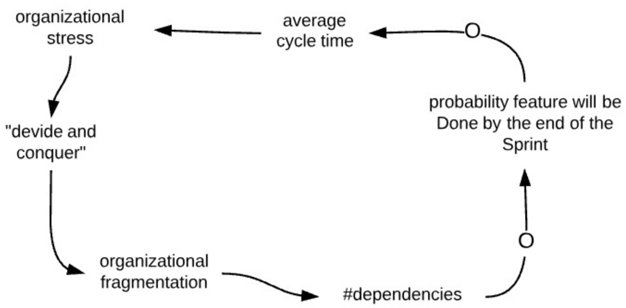
Posted on July 11, 2018

If you worked in large organizations you have probably heard about the term "dependencies". I am convinced that dependencies need to be eliminated, not managed. With a help of system diagrams in this article, I will uncover the main reasons why Scrum Teams suffer from dependencies, how they impact organizational agility, and what the fundamental solutions to this issue are.

## How dependences inhibit Teams progress

The more dependencies, the less chances the feature will be done by the end of the Sprint. Thus, the more time it takes for the feature on average to go from Product Backlog queue to the market (cycle time). As a result, agility is reduced because the organization is unable to deliver potential value to the market quickly. This causes organizational stress.

A typical management response to organizational stress is "divide and conquer". For instance, if there is a problem with the quality, let's create a separate department “quality control” with set of its own KPIs. Creating new functions, units, component teams and coordination roles, managers strengthen the fragmentation of the organization. More fragmentation leads to even more dependencies.


organizational stress
average cycle time
O
probability feature will be Done by the end of the Sprint
"devide and conquer"
O
organizational fragmentation
#dependencies

High average cycle time makes the organization less agile. But Scrum Teams should not have any dependencies!

> *Cross-functional teams have all competencies needed to accomplish the work without depending on others not part of the team. The team model in Scrum is designed to optimize flexibility, creativity, and productivity.*
>
> Scrum Guide 2017

## Why dependencies exist

From my long experience working with large organizations I see a few reasons why the dependences thrive:

- Imperfect organizational design based on component teams ( "bus team", "analytics team", "Android Team", “integration team”). It causes intensive fragmentation.
- Incomplete cross-functionality (lack of one or more skills).
- Unreasonable complicated architectural design ( f.e. "there are 256 systems in our organization"), which inhibits creation of cross-component and cross-functional Scrum Teams.

# How to get rid of dependences

The dependencies issue could be solved in two ways: with quick fixes or fundamental long-term solutions.

The quick fix is to visualize and manage the dependencies. E.g. creating additional coordination roles  or using specific techniques ("ropes on the boards"). Yes, it somehow helps to survive and continue the movement. You become an artist of the visualization and dependency management. Your work looks similar to this:

The fundamental solution is to eliminate the fundamental issues completely by:

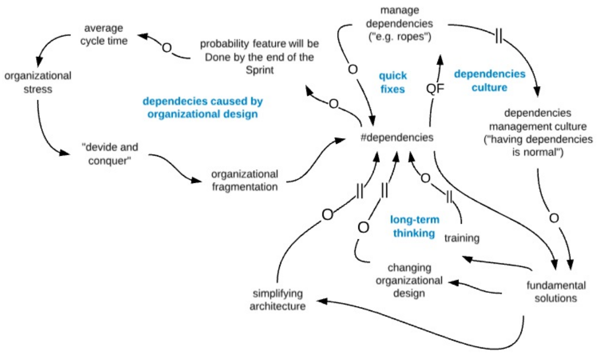
- training people
- making the complicated architecture simpler, reducing the number of components
- changing organizational design and forming cross-component cross-functional Scrum Teams (Feature Teams)

In this case the board for the scaled Scrum could look much better (no dependencies):

Feature Teams do not need the ropes because there are no dependencies or they are trivial.

Let’s get back to the system diagramming. On one hand, we have a cycle explaining the rise of dependencies, on the other hand it is a quick fix of the problem, then a fundamental solution cycle and the final cycle of forming a culture of managing dependencies when time passes.

*The bad news is that a strong culture of “managing dependencies” will hinder the implementation of the fundamental solutions.*


manage dependencies ("e.g. ropes")
average cycle time
probability feature will be Done by the end of the Sprint
organizational stress
quick fixes
QF
dependencies culture
dependecies caused by organizational design
#dependencies
dependencies management culture ("having dependencies is normal")
"devide and conquer"
organizational fragmentation
long-term thinking
training
changing organizational design
fundamental solutions
simplifying architecture
O
O
O
O
O
O
O
||
||
||
||

1. The more dependencies the less agile organization becomes.

2. Dependencies thrive because of the unnecessarily complex architecture, lack of skills, and suboptimal organizational design (component teams).

3. Creating additional roles and using dependencies management practices do not eliminate the fundamental issues.

4. Fundamental solutions are simplification of the architecture, Feature Teams and, training people.

Do you manage dependences or want to eliminate them?

**Scrum ON!**

***

To read this article online with embedded hypertext links, go here:

**https://www.scrum.org/resources/blog/eliminate-dependencies-dont-manage-them**

# About Illia Pavlichenko

**Illia** works on large scale Agile transformations worldwide as a management consultant and product development expert. He has introduced Agile thinking at various companies ranging from high-tech, financials to internet companies. He is also a Certified LeSS Trainer and a Professional Scrum Trainer™.

***

# Sustainable Pace in Product Management

By Roman Pichler

Posted on November 6, 2018

Working in product management is rewarding but demanding. As product people, we have a large set of diverse responsibilities, which often translates into a high workload. But continuously working too hard carries the risk of becoming chronically tired and stressed and sacrificing our health. This article discusses techniques that help you achieve a healthy, sustainable pace and avoid the danger of being constantly overworked.

## What is Sustainable Pace?

Sustainable pace is an important agile principle. The Agile Manifesto defines it in the following way: “The sponsors, developers, and users should be able to maintain a constant pace indefinitely.” The goal is to create a healthy work environment and avoid that people are routinely overworked, lose their creativity, make mistakes, and eventually sacrifice their health. A framework like Scrum offers specific techniques that ensure sustainable pace—unfortunately, only for development team members and not for product people.

But sustainable pace is equally important for you, the person in charge of the product. You have a demanding job with a range of diverse responsibilities. These include interviewing users, working on the product roadmap, updating the product backlog, engaging with the stakeholders, and working with the development team, to name just a few. As all these duties compete for your time and attention, it is all too easy to do too much, work too hard, and exhaust yourself—which is neither good for you, nor for your product.

---

## Say No

If you find that you are overworked and struggle to cope with your workload, then start by reflecting on the tasks you carry out. Are all of these part of your actual job? I find that product people often take on responsibilities that belong to other roles thereby making a demanding job even harder. A common example is looking after the development team. While it’s great to care about the team, facilitating effective collaboration within this group is not your responsibility. That’s the job of the Scrum Master.

I know that some product people don’t have a Scrum Master working with them or that the Scrum Master’s work is not effective—the individual might be too stretched or not adequately qualified. But if that’s the case for you, then I recommend addressing the issue rather than covering another person’s job. The latter will stabilise an ineffective setup and cause you to be overworked or neglect some of your core duties, neither of which is desirable.

Therefore, focus on your actual job—making or keeping the product successful. Do not take on additional responsibilities like improving the development process, leading the dev team, making UX design decisions, or creating a marketing strategy. That’s the responsibility of the Scrum Master, development team, and marketing stakeholder respectively, not yours. Have the courage to say no even if it’s difficult: There is no lasting benefit in you becoming the general dogsbody for the product.

---

## Be Proactive

Next, minimise the amount of unplanned work and firefighting you encounter. I find that many product people are so busy with urgent tactical work, such as refining user stories, working with the development team, or answering a support request, that they neglect important strategic tasks like regularly assessing if the product strategy is still working. This can cause nasty surprises like a competitor leapfrogging you, which then leads to more, unplanned work, as you desperately try to catch up with the competition.

Consequently, make enough time for strategic work. Regularly assess how your product is doing and how effective your current product strategy is, for example, by holding collaborative strategy reviews, as I describe in more details in the article “Establishing an Effective Product Strategy Process”. This will allow you to play a proactive game and be responsive rather than having to react to surprises.

## Share the Work

If you find that you are simply too busy to regularly strategize and cannot let go of any responsibilities, then consider sharing your workload with development team, stakeholders, and other product people.

### With the Development Team

If you spend a lot of time working on the product backlog trying to create perfectly crafted user stories or if you have to answer plenty of questions during the sprints, then you are probably not effectively sharing the product backlog work. Managing the product backlog should be a collaborative effort. The development team members should actively participate in the backlog work, discover, capture and update stories together with you, and help you prioritise the product backlog. This leads to better product backlog items, reduces the amount of questions you have to answer in the sprint, and frees up your time.

Additionally, you are often able to delegate (some of) the refinement work to the development team—assuming that the team has acquired enough knowledge about the users and product and that you trust the individuals to make the right decisions. This will further reduce your workload and enable you to spend more time on important strategic tasks.

### With the Stakeholders

Some product people I have met spend a significant amount of their time on “politics”: negotiating deals, convincing people, selling ideas to important stakeholders. Consequently, stakeholder management can feel like herding cats. While you will always encounter stakeholder challenges, I find that a lot of time spent with negotiation, persuasion, and re-alignment activities can be saved when you embrace a participatory decision-making process.

The idea is simple: Involve the key stakeholders—together with development team members—in important product decisions, for instance, by using consensus or product person decides after discussion as the decision rule. This will increase people’s buy-in and reduce the need for continued re-alignment meetings or possibly some crisis management if you find out that people did not implement an important decision. (You can find out more about participatory decision-making in my article “Use Decision Rules to Make Better Decisions”.)

### With Other Product People

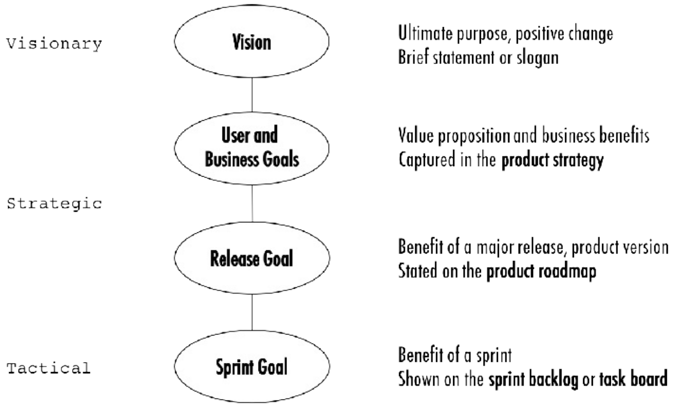
If the previous two measures are not appropriate, then consider how big your product is and how many development teams you work with. As a rule of thumb, if there are more than three teams, then you may have to involve additional product people to help you manage the product. You could have, for example, one person in charge of the overall product and additional people who look after product features, as the picture below illustrates.

Alternatively, you may want to consider breaking up your product, for example, by unbundling one or more features and launching them as a separate product like Facebook did with the Messenger app in 2014. Both options will reduce your workload and make it easier to achieve sustainable pace. (Please refer to my article “Scaling the Product Owner Role” for more information on how to jointly manage a product.)

## Prioritise

Many years ago, I was discussing a lengthy requirements document with the product manager in charge of a healthcare product. As there was too much work to do, I suggested prioritising the requirements. I’ll never forget the look I received and the answer the individual gave me. She said: “That’s impossible. They are all important!” The issue is, of course, that without the ability to prioritise, we don’t know when to say yes and when to say no. Consequently, we are likely to take on too much work and trying to accomplish too many things at once.

If you find it hard to prioritise—be it the order in which items should be delivered or if you should accept a feature request—then you will benefit from establishing clear and shared goals. As described in my article “Leading through Shared Goals”, I like to work cascading goals that form the chain shown in the picture below.


Visionary
Vision
Ultimate purpose, positive change
Brief statement or slogan
User and Business Goals
Value proposition and business benefits
Captured in the product strategy
Strategic
Release Goal
Benefit of a major release, product version
Stated on the product roadmap
Tactical
Sprint Goal
Benefit of a sprint
Shown on the sprint backlog or task board

The goals in the picture above are systematically linked and constitute a hierarchy with the vision at the top and the sprint goal at the bottom. With the right goals in place, you are able to assess if you should add a new feature, for example. Here is how it works: If the feature helps you reach your current release goal, you should probably take it on. If it doesn’t, then include it in the product roadmap—assuming it serves a user or business goal stated in the product strategy. If that’s not the case, then kindly but firmly say no to it. In the story above, the product manager would have benefited from having a clear and agreed release goal. This would have allowed her to order the requirements by considering how important each one was for meeting this goal.

Therefore, ensure you have meaningful and agreed goals available that help you prioritise the work. This will not only help you make the right decisions, it will also reduce the risk of being overworked and help create a sustainable pace.

***

To read this article online with embedded hypertext links, go here:

**https://www.romanpichler.com/blog/sustainable-pace-product-management/**

# About Roman Pichler

**Roman** Pichler is a product management expert specialised in digital products. He has more than 15 years experience in teaching product managers and product owners, and in helping companies build successful product management organisations. Roman is the author of three books, including *Strategize* and *Agile Product Management with Scrum*, and he writes a popular blog for product professionals. Find out more at https://www.romanpichler.com.

# Looking at Agile Coaching and Sports Coaching

By Allison Pollard

Posted on February 20, 2018

Photo by Greg Goebel

It’s common for agile coaches to be compared to sports coaches. Coaches fill a well-recognized role in sports, and many people have had some real experience on a sports team with a coach. In fact, I often see Scrum Masters or Agile Coaches being represented in diagrams as the person with a hat and a whistle, suggesting the sports coach metaphor. It’s a great comparison to explain how a Scrum Master or Agile Coach is typically outside the “software development game”—just as a sports coach is not scoring the points, this agile role is not hands-on in creating the product.

As I think about my experience on a sports team as a kid, I remember my softball coach showing me how to hold a bat, how to position my feet, and how to swing the bat in order to hit a ball. It was awkward and mechanical at first. And that’s where the metaphor of a sports coach starts to bother me. We could spend a lot of time teaching a team the mechanics of every agile event or artifact—it would be overwhelming for the team to absorb and apply. One aspect of agile coaching is teaching. There’s also mentoring, facilitating, and coaching. Many people say that an agile coach initially teaches the team, and as they mature, the coach moves into more of a facilitating or coaching stance. That is one way a coach can work. It’s possible to facilitate or coach much earlier in a team’s agile adoption without all of the upfront mechanics lessons. Sir John Whitmore illustrates the differences between coaching and instruction in a video about tennis.

Rather than spend more time upfront teaching and explaining how something is to be done step-by-step, a coach can help a team explore their experience of doing something. The coach deepens the team’s awareness of what they’re doing and how to do it differently. In doing so, the team owns the way something is done from the beginning, learns to recognize what is working and what is not, and is engaged in thinking of options to improve. The team is doing and learning from doing. The inspect and adapt cycle that we encourage in teams is introduced from the beginning. Imagine how a team might embrace agile then!

***

To read this article online with embedded hypertext links, go here:

**https://www.allisonpollard.com/blog/2018/2/20/looking-at-agile-coaching-and-sports-coaching**

# About Allison Pollard

**Allison** Pollard helps people discover their agile instincts and develop their coaching abilities. As an agile coach with Improving in Dallas, Allison enjoys mentoring others to become great Scrum Masters and fostering communities that provide sustainability for agile transformations. In her experience, applying agile methods improves delivery, strengthens relationships, and builds trust between business and IT. Allison is also a Certified Professional Co-Active Coach, a foodie, and proud glasses wearer. Visit her blog at www.allisonpollard.com to connect further.

# Are You Asking The Right Questions?

By Lizet Pollen

Posted on June 25, 2018

One of the things I enjoy about coaching is the large variety of challenges, goals and desires clients present. Some clients know exactly what they want, but they're unsure how to get there, or they need an accountability partner to follow through. Others have so much going on, in their lives or at work, that they don't even know where to start.

Regardless of what a client presents at first, it always comes down to this:

***People hire a coach because they want some kind of CHANGE, in one or more areas of their life.***

## Deficit or Possibility?

There are two lenses through which you can look at **CHANGE**: the deficit lens and the possibilities lens.

1. **The lens that focuses on deficit.** If you look through this lens you zoom in on what's wrong, broken or not working and you take this negative perspective as your starting point for change.

You would look at your life and notice things like:

- *I'm stressed and overwhelmed with everything I have to do*
- *Others keep crossing my boundaries, because I fear not being liked*
- *My co-workers perform better than me*

You're probably experiencing distress and you want **change** in your life:

- *I don't want to get burned-out*
- *I don't want others to keep walking all over me and disrespect me*
- *I don't want be stuck in this job forever*

Looking at your life through this lens evokes negative emotional states, like anxiety, shame, anger or depression. When those emotions are evoked the primitive part of the brain is activated and you'll automatically switch to survival mode. Not only does this cause us to disconnect from other people around us (focus on 'me' instead of 'we'), it also causes tunnel vision, preventing us from seeing the big picture and all the resources and solutions that might be out there.

***Not a great place to be in when you want to create CHANGE in your life, right?***

2. **The lens that focuses on possibilities**. If you look through this lens you zoom in on the good, the better and the possible and you take this positive perspective as your starting point for change.

You would look at your life and notice things like:

- *I'm focused and content after taking a break*
- *I sometimes say 'no' to people that are close to me*
- *I have the skills and the talent to develop in this job if I'm patient and put effort in*

Looking at your life through this lens sparks hope and unleashes confidence and self-efficacy. When those emotional states are evoked the more evolved part of the brain (neo-cortex) is activated and you'll switch to **thriving mode**!

***Do you see how replacing the deficit lens with the possibilities lens pays off when you want to create CHANGE in your life?***

## The Flip

This quote by David Cooperrider, sums it up nicely: "Change starts with the first question you ask".

So, rather than asking yourself what you don't want (anymore), ask yourself what's already going well in your life, what you want to grow, what you want more of!

This flip might look something like this:

- *I don't want to get burned-out* ⇒ *I want to be more focused and content.*
- *I don't want others to keep walking all over me and disrespect me* ⇒ *I want to be more confident and assertive.*
- *I don't want be stuck in this job forever* ⇒ *I want to grow and develop in this job.*

***What do you want more of in your life?***

***

To read this article online with embedded hypertext links, go here:

**https://www.impowered.net/single-post/2018/06/25/Are-You-Asking-The-Right-Questions**

# About Lizet Pollen

**Lizet** Pollen is a certified coach, trainer and psychologist. She’s the founder of ImPowered Coaching & Training(www.impowered.net). It’s her passion to help individuals, teams and businesses to Thrive and Excel, by showing them how to boost performance & results and improve their lives across the board. She helps to create awareness, gain confidence, move beyond mental and/or emotional 'roadblocks', create new habits, and ultimately to integrate the desired changes into their work and/or life.

ImPowered offers Life & Career Coaching, as well as (custom made) Training Modules & Programs. A strengths-based approach is combined with evidence-based insights & strategies from the field of (Positive) Psychology.

Lizet was born and raised in the Netherlands, she’s married to an Irish men and mother of 2 girls, and happily living in the state of Texas. Some of the things she loves are travelling, photography, wining & dining, the outdoors and reading.

Connect with Lizet on Facebook (https://www.facebook.com/impoweredcoaching/), Instagram (https://www.instagram.com/impoweredcoaching/) or Linkedin (https://www.linkedin.com/in/lizetpollen/)

# Agile? DevOps? You won't get there with Fear in your workplace!

By Dana Pylayeva

Posted on November 23, 2018

Fear in the Workplace game - fun way to talk about the serious stuff.

### Take your first step towards a fearless organization. And it doesn't have to mean "resign from the one you are in right now".

## Illusion of success

Psychological safety has been a hot topic at Agile conferences and meetups ever since Google published the results of their Project Aristotle. DevOps practitioners have known for a while about the third principle underpinning DevOps - Culture of Continual Experimentation and Learning.

What do you think? **Is there a chance that the Third Way of DevOps and the Psychological Safety might be related?**

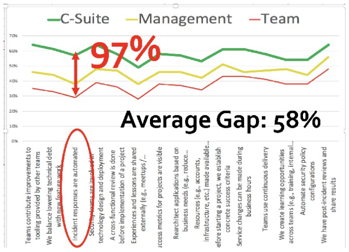
I've asked this questions for the first time at Agile + DevOps East 2018 conference during my lightning keynote. To my surprise, only a few hands went up, confirming the connection. Since then I've been running this "informal survey" with every new audience and the results have been consistent:

**A very few people recognize a link between the two.**

My interpretation of the disconnect is that this is a by-product of companies starting their DevOps journey with automation initiatives only. They hire DevOps engineers, create new DevOps teams or even simply re-brand their Sys Admins and do nothing major to truly break down the silos between Development and Operations. People in those organizations are likely to continue to operate from the place of fear and be unable to experience all the benefits that come from embracing the DevOps culture. What's even worse, as Amy C. Edmondson noted in her latest book "The Fearless Organization":

> *"A lack of psychological safety can create an illusion of success that eventually turns into serious business failures."*

This "illusion of success" of DevOps practices has even emerged in 2018 State of DevOps report by Puppet and Splunk. During the data gathering process of this report, C-level executives, management and team members were asked a series of questions, fifteen of which were related to penetration levels of various DevOps practices in their organizations.



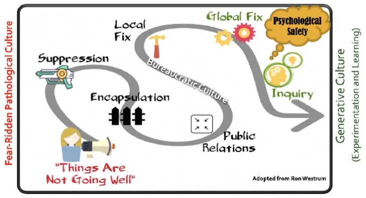
Based on data collected by Puppet for State of DevOps Report 2018

I've generated a simple graph based on those responses and the gap between each group's perception became even more pronounced! On average C-level executives reported 58% better penetration levels than those observed by teams in the trenches, with the astounding 97% variance in responses to "Incident responses are automated" statement.

When team members don't feel safe to share with the C-levels the true status of DevOps adoption progress, it is highly unlikely they will be willing to speak up about ambiguous threats, ask questions or share their opinions, when they don't align with those of the top level management. We've seen a number of cases in the recent past where this led to catastrophic failures and significant financial losses for the organizations. From Volkswagen emissions scandal to Wells Fargo account fraud, culture of fear silenced anyone who could ask questions, push back on unreasonable demands or criticize unethical workarounds.

## What Ron Westrum knew about fear

Fear Cripples Information Flow



Adopted from Ron Westrum

Ron Westrum classifies organizational cultures into Pathological (fear-ridden), Bureaucratic and Generative cultures. DevOps and Agile thrive in Generative culture while can potentially trigger new fears and even face a strong resistance in Bureaucratic or Pathological cultures. It doesn't have to be that way! Yes, it will certainly "take a village" to change the organizational culture. Yes, it will require a strong support from the top. However, I believe that each and every one of us has the power of taking initial steps.

**Start a conversation about the impact of fear and practice recognizing its symptoms in your workplace!**

If you've been following my work, you know that I love designing and using Agile games with my clients. The one I've been experimenting with recently is Fear in the Workplace. The game includes common fears and fear symptoms collected in my various client engagements, selected from extensive safety literature and identified in coaching conversations.

In my experiments, using Fear cards to help facilitate these conversations have led to more openness amongst team members. Somehow, pointing to a silly monster image made it easier to externalize one's feelings, to step back and have productive conversations. In one of the organizations, I even brought in the Fear deck to facilitate a conversation between a new remote manager and her co-located team. That's how we discovered that the manager had a "Fear of unhappy team". Every card in this deck has a story and enables rich conversations. The deck includes also "ClassifyMe" cards - typical scenarios and things people say in the workplaces affected by fear. While initially collected in specific organizations, over time these scenarios have been proven to be common in other workplaces as well.

## Fun ways to start the fear conversation

## 1. Simple option - for competitive teams

One of the easiest ways to start the conversations about Fear is by playing a **simple game** with your team (based on the "Cards Against Humanity" dynamic):

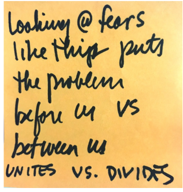
1. Deal the "Fear and Fear Symptoms" cards to all players. Place the "ClassifyMe" cards in the middle (face down).
2. Have one player (the Reader) read a ClassifyMe card and others select the best matching Fear or a Symptom card from their hands.
3. Have a conversation about each match and have the Reader pick the best match for their ClassifyMe card.
4. Player with the best match wins the ClassifyMe card. Discard all other fear cards proposed in this round. Continue the game, rotating a role of the Reader.
5. Do a debrief with the group after the game is over. Help them reflect on the connections they've made with their own workplace.

This version is the best suited when you have a small group and at least 15 min of time.

## 2. Simple collaborative option

Many teams will opt for a collaborative version of this game for two main reasons:

First of all, there are always more than one fear or symptoms that can be traced to a selected "Classify Me" card. In fact, a conversation becomes richer, as team proceeds to building a full story. Cards codify variety of fear/fear symptoms and help the team explore their causation. In these discussion, fighting against "fear monsters" team comes together, develops empathy and starts to open up for deeper conversations. As one of the participants noted in his feedback:



Looking @ fears
like this puts
the problem
before us vs
between us
UNITES VS. DIVIDES

> *"Looking at fears like this puts the problem before us vs. between us. Unites vs. divides."*

Second reason for the collaborative game preference has to do with the fact that this version makes all fears offered by each player valid. Unlike the competitive version, players don't discard the Fear/Symptoms cards. Instead, each player adds a card and their story to the entire pictures reconstructed by the team.

With a more advanced group you can use these cards to run LeanCoffee style discussions by first dot-voting on Fear and Fear Symptoms cards and then creating a "discussion backlog" for your time-boxed conversation. This version becomes even more powerful when combined with "Safety in Workplace" set - a collection of "Psychological safety enhancers" that can be used at an individual level, team level and a leadership level.

## 3. Large group/short timebox option

What if your group is large and the timebox is short? That was exactly my challenge in the lightning keynote at Agile + DevOps East 2018! I used this deck to lead **400 people in 5 min** through their first discussion about fears. Here is what worked for me:

1. Distribute a fear/fear symptom card to every other participant.
2. Briefly introduce the impact of fear on our ability to innovate and connect 3rd way of DevOps with Psychological Safety (2 min)
3. Invite them to form pairs or triads and ask each other three questions (2min):

- What is your name?
- What is your role?
- How often have you observed this fear in your workplace?

4. In the last minute invite them to continue this conversation with their partners from the exercise or in their workplace.

Even this very short introduction to the Fear in the Workplace topic helps people understand that they don't need for suffer silently in their toxic cultures. As another participant put it, it is a

***Fun game that makes you think***

## 4. A Dash of Liberating Structures (advanced option)

This version was born out of my latest fascination with Liberating Structures(LS). Once you have a team warmed up with Fear in the Workplace game, they might be ready to dive deeper into their own workplace fears. Here is when you can introduce the "Drawing your monster" LS "under development"(based on Lynda Barry's work). I first experienced this LS when shadowing Nancy White in her LS Immersion workshop and immediately made the connection!

When I run it with my teams or at a conference, I bring a stack of coloured paper to make this exercise more cheerful and ask participants to:

1. Make a list of things that worry them the most in their workplace.
2. Circle the 4 scariest, most alarming things.
3. Flip the sheet to another side, and draw 4 different shapes: a circle, a square, a spiky shape and a random shape. These become the bodies of their monsters. Next they add all sort of features - legs, fins, eyes and map their fears from step 2 to their monster drawings.

These is where the most amazing things happen. When listing their fears, people feel apprehensive at first: "Will we have to share these with others?". By the time they are done drawing their monsters, they are laughing, talking and they WANT to share these with others. Here is where the "Monster Walk" comes in!

They are invited to take their monsters for a walk, show them to others and ask for their advice on how to deal with these monsters. They come out of these conversations energized, empowered, having built stronger connections with their team members. And their fears? They become less scary after this walk.

I hope my experiments inspired you to take a first step in your workplace. Give it a try! Bring Fear in the Workplace game to your teams and your leadership, experience the power of Liberating Structures. Above all, please share your experience with taking steps towards **building a fearless organization. Yours!**

***

To read this article online with embedded hypertext links, go here:

**https://www.linkedin.com/pulse/agile-devops-you-wont-get-fear-your-workplace-dana-pylayeva/**

# About Dana Pylayeva

**Dana** Pylayeva is an independent Agile/DevOps coach and trainer with eighteen years of experience in IT. She works with organizations in the US, Canada, Ireland, Japan, and Russia and speaks internationally about DevOps, Agile, and psychological safety. Dana is a game designer, the author of "*Introduction to DevOps with LEGO, Chocolate, and Scrum Game*", "*Self-selection game*" and "*Fear in the Workplace*" game. She is a certified Training from the Back of the Room trainer, Open Space facilitator, CSM, CSP, CSPO, ICP-ACC and Certified LeSS Practitioner. An enthusiastic collaborator, she builds communities, organizes conferences, and enjoys connecting with agilists from around the world. Dana is the founder of the Big Apple Scrum Day conference in NYC, organizer of NYC Liberating Structures meetup, a program co-chair of the Agile2019 conference and a member of NAGT for Global Scrum Gathering SGNYC20. Learn more about Dana on her company's website and reach out at dana@agileplayconsulting.com.

# Where There's No Friction There's Flow: The Four Pillars of the Fearless Tribe

By Garry Ridge

Posted on December 11, 2018

Multiple leadership styles and philosophies may bring any CEO the desired results of market performance, both on Main and Wall Streets. But the options get fewer and fewer as the CEO puts *sustainable* performance on the agenda. Throw in the ideal of a cultural sense of ease, passion, commitment that would permeate all the stakeholders of the company's activities throughout time and throughout the world, and you have only one leadership style option that will work:

The CEO must be completely dedicated to creating and sustaining a culture where all participants are free to focus, innovate, speak up, be themselves, make mistakes, be happy at work, and wholeheartedly align their personal purpose with the company's purpose. This is the requirement, even during those rare periods that demand a certain amount of personal sacrifice and faith in the leadership.

How does this kind of commitment to a positive culture at a macro level over time bring sustainability to the CEO's expectations for the company's success year over year, decade over decade? This commitment to a cultural consistency creates a community of interdependent and mutually supportive people. At WD-40 Company we call this community our *tribe*. Tribe members bring to their jobs, careers, relationships and lives the following critical attributes:

- Fearlessness
- Resilience
- Trust
- Drive for ongoing learning and teaching
- Enthusiastic engagement while working, lavish celebration at the success of initiatives
- Selflessness (to a healthy, reasonable extent) in service of the community priorities and fellow tribe members
- Clarity of mission
- Dedication to purpose
- Joyful tapestry of unity and belonging that crosses oceans, spans time zones, melts national boundaries, and embraces the singular human spirit that makes up the WD-40 Company tribe.

What emerges from this culture – beyond the obviously desirable day-to-day experience of working with people we care about – is that truly enviable performance history on every key metric that drives WD-40 Company's success story. Wherever I go, anywhere in the world, I am asked, "How do you do that?"

To coin a phrase: The process is simple. But it's not easy. It requires commitment to high ideals at times when less-committed leaders might fold in pursuit of one more percentage point in profit, or when they must choose among an environmentally responsible ingredient, profit margin, or affordable price point for the consumer.

The business literature is filled with many fine stories of efforts that CEOs make to recruit their employees in the mission of creating a business success story. From the long-term, sincere efforts to the short-term gimmicky attempts, there is no shortage of ideas you can try in your own company. But from what I have found over time is that nothing will hold up in any kind of meaningful way until four fundamentals are in place.

I call them the **Four Pillars of the Fearless Tribe**. Imagine, for a moment, the sub-basement of a spectacular skyscraper. It's not so glamorous down there, to be sure. It's mostly bland concrete, specially formulated and poured to take on the weight of all that is above – all those details and amenities selected to please, inspire, communicate, transform, enlighten, and ignite the imagination of all who interact with the building and its occupants.

But you are unlikely to see any of that in the sub-basement. What you do see are massive, unadorned pillars. Considering the accumulated weight of all that they hold up, there are surprisingly few. In fact, the fewer the pillars, the better the engineering.

Such is also the case for the Four Pillars of the Fearless Tribe. As we have engineered the WD-40 Company's cultural structure over time, we found that these four pillars hold up the entire experience of what it means to do business within the company. Each pillar is critical. They are:

- Care
- Candor
- Accountability
- Responsibility

These are the essential structural supports, providing a distributed foundation under emotional weight that organizational psychologists refer to as *psychological safety* inside the workplace. In this document, I will first tell the story of WD-40 Company's success in metrics form. Then I will introduce to you the basic concepts of psychological safety as the academics are reporting on it today. And finally, I will describe our own four pillars, as we have identified them to be essential to holding up the culture that makes the WD-40 Company the joyful, successful tribe that it is today.

## What If All Your Employees Loved Coming to Work Every Day?

To explore our tribal culture as the "secret sauce" of our successes, the first thing to do is define what we at WD-40 Company mean by the word "tribe." Simon Sinek talks about a "circle of safety," in which all the participants are collected and protected by a defining mutual agreement of values, practices, mission, purpose, and ways of doing things. There is an ongoing sense of belonging that is as consistent and dependable in the environment as the air we breathe. Once we are accepted into the group, we trust each other. We share knowledge freely and openly. We assume the best of each other's intentions at all times. We sacrifice for each other. We celebrate with each other. We honor our contributions as individuals without losing sight of the valuable, positive impact on the entire community. While no organization can achieve this state constantly or without being tested, our objective is to work toward achieving this condition at least 95% of the time.

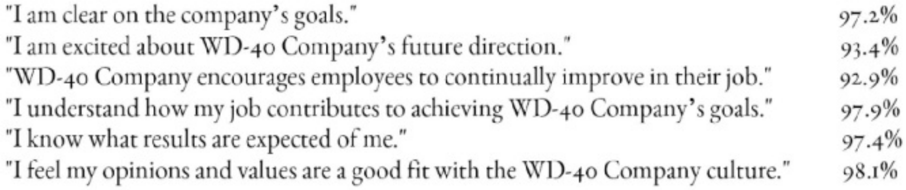
In his book, *Tribe: On Homecoming and Belonging*, Sebastian Junger puts it this way: "The earliest and most basic definition of community – of tribe – would be the group of people that you both help feed and help defend."

Taking the effort to define and create a tribe in your company has obvious significant and unmistakable rewards. Dave Logan, John King, and Halee Fischer-Wright specify them beautifully in their book, *Tribal Leadership: Leveraging Natural Groups to Build a Thriving Organization*:

- People collaborate and work toward a noble cause, propelled from their values.
- Fear and stress go down as the "interpersonal friction" of working together decreases.
- The entire tribe shifts from resisting leadership to seeking it out.
- People seek employment in the company and stay, taking the company a long way toward winning the war for talent.

- People’s engagement in work increases, and they go from “quit on the job but still on the payroll” to fully participating.
- Organizational learning becomes effortless, with the tribe actively teaching its members the latest thinking and practices.
- People’s overall health statistics improve. Injury rates and sick days go down.
- Setting and implementing a successful competitive strategy becomes stunningly easy as people’s aspirations, knowledge of the market, and creativity are unlocked and shared.
- People report feeling more alive and having more fun.

How has our commitment to a tribal culture manifested itself in the experience of working at WD-40 Company and our market performance? That story is best told by the results of our 2018 Employee Engagement Index. Below are just some of the 26 queries that we have identified as key to our performance both in the marketplace globally and our own noble cause of making sure we are providing a workplace culture where our employees feel safe, supported, appreciated, inspired, innovative, curious, and optimistic. About 94% of global employees completed this survey, in seven languages:



| | |
|---|---|
| "I am clear on the company’s goals." | 97.2% |
| "I am excited about WD-40 Company’s future direction." | 93.4% |
| "WD-40 Company encourages employees to continually improve in their job." | 92.9% |
| "I understand how my job contributes to achieving WD-40 Company’s goals." | 97.9% |
| "I know what results are expected of me." | 97.4% |
| "I feel my opinions and values are a good fit with the WD-40 Company culture." | 98.1% |

And, my personal favorite:

**“I love to tell people that I work for WD-40 Company.” 99.0%**

All these answers are experience-based. The way our tribe members experience their time at WD-40 Company shapes their perceptions of the company, and whether they feel it is a good place for them to invest their time, talents, and passion. Providing those experiences is, to all of us at WD-40 Company, a sacred responsibility for multiple reasons. We recognize that people spend the majority of their waking hours at work. So why shouldn’t they feel fully fulfilled and supported by people they know, like, trust and respect? Don’t we all deserve that daily expectation as part of our natural inheritance as humans in this world? And they bring those positive feelings home to their families at the end of the day. So we know that our tribe members’ positive feelings about the work they do contribute to the optimism that their family members carry with them into their own futures.

How does this tribal engagement manifest in company performance? There is abundant literature showing the linkage between high engagement scores and company performance across companies and industries. So I’ll just focus on what we’re experiencing at WD-40 Company:

Over the past 20 years that we have committed to these foundations, our sales have quadrupled. Our market cap has increased from $250 million to nearly $2.5 billion. And during these last two decades, our annual compounded growth rate of total shareholder return is 15%.

The proof of the pudding, as they say, is in the eating. Those numbers represent a whole lot of product. But our second most important value is, “We create positive lasting memories in all of our relationships.” Engagement, and thus personal investment in the organization, stems from positive experiences within the tribe, which yields the applause of financial performance.

## Psychological Safety – A Brief Introduction

*In this dynamic environment, successful organizations need to be managed as complex adaptive systems rather than as intricate controlled machines.*

Amy C. Edmondson,

### *Teaming: How Organizations Learn, Innovate, and Compete in the Knowledge Economy*

It’s been almost a century since productivity experts turned their focus on how companies can optimize production by improving the human aspect of the organizational systems. In the earliest decades, their attention was on how humans and machines could interact better, incremental improvement by incremental improvement – with humans standing in the service of the machines. Then, predictably, as the Knowledge Economy took predominance over the manufacturing sector, the focus was turned to how individual employees could perform better as separate functional entities – their workstations being primarily between their ears – generating each company’s competitive edge through innovative, original thinking.

Consequently, over time, organizational psychologists and corporate leaders are coming to fully realize the critical role that mental health plays in a company’s performance. As the Knowledge Economy evolves, we leaders are beginning to acknowledge – even, for some of us, embrace – our roles in creating a cultural experience where our people are free to contribute their best because they feel safe. As Maslow's Hierarchy of Needs has proven, without a sense of safety (survival, family preservation), it's impossible to think long term and to stay engaged. After those basic, hygienic needs are met, people then need to feel that they belong to a welcoming, non-judging tribe, where they can count on the support of others. Then they can focus on their performance, secure in the knowledge that their community of colleagues – their tribe – wish them the very best. And the feeling is reciprocal.

This is called *psychological safety*, an expression first introduced by the organizational culture pioneer Edgar Schein. Amy Edmondson, the Novartis Professor of Leadership at Harvard Business School, has subsequently developed this concept to encompass the entire team experience. Edmondson’s research encourages a shift in focus from production (which she calls “organizing to execute”) to a “new way of working that supports collaboration, innovation, and organizational learning.”

“Learning in today’s organizations involves what’s called *reciprocal interdependence*, where back-and-forth communications are essential to getting the work done,” Edmondson writes in *Teaming*, outlining workplace conditions where a psychologically safe culture is essential:

- When the work requires people to juggle multiple objectives with minimal oversight.
- When people must be able to shift from one situation to another while maintaining high levels of communication and tight coordination.
- When it is helpful to integrate perspectives from different disciplines.
- When collaborating across dispersed locations.
- When pre-planned coordination is impossible or unrealistic due to the changing nature of the work.
- When complex information must be processed, synthesized, and put to good use quickly.

What culture in today’s global business environment does *not*have any of the elements that require team members to work together freely, generously, and fearlessly? And yet, there are so many opportunities inside every business, within every business day, where the best of intentions, the best of ideas can go to smash because of a misunderstanding or betrayed trust.

Edmondson adds, “The knowledge-based economy only works well when it restores workers on all levels to self-respecting, self-determining adulthood.”

For this to be effective, they need to feel safe – not only physically safe but emotionally secure to focus on their work and bring new ideas to the table without fear of retribution. Just as we at WD-40 Company celebrate what we call the *learning moment*, Edmondson calls leaders’ attention to the need for companies to emphasize their value on learning without reprisal as a key component to the psychologically safe culture.

“This calls for workers who know how to experiment, how to think on their feet, how to work in the absence of rules, and how to adapt quickly,” she writes. “Knowledge, changing quickly within disciplines, becomes even messier and more uncertain when integrated across disciplines…to get things done in the new workplace. Creating an appropriate environment for teaming and learning requires different management skills and expectations from those required in a repetitive task environment…today’s managers need employees to be problem solvers and experimenters, not mere conformers.”

To cultivate a culture of these confident, independent, interdependent team members, leaders are increasingly recognizing the need for their entire community of talent to be integrated by the same sets of values and expectations. When those are in place, they are free to focus, create, and bring your company to the fore in its competitiveness and ability to attract and retain the very best talent to continue the march toward the future.

At WD-40 Company, this culture starts with those Four Pillars of the Fearless Tribe.

# The Four Pillars Explained

*Knowledge is only rumor until it lives in the bones.*

Asaro tribe in Papua New Guinea,
as quoted by Brene Brown, *Dare to Lead*

The success stories of other companies are especially interesting when they carry indicators of how we can transform their examples into actionable insights to apply to our own organizations. It’s one thing to know the success metrics of highly admired companies such as WD-40, but the real value is understanding the ways you can duplicate WD-40 Company’s approach to the extent that’s appropriate to your business.

Your culture is different, as it should be. So copying WD-40 Company’s example in all its details – as you might a cookbook recipe – could set you up for failure. However, the foundational pillars that I introduced above are duplicable anywhere. No matter your industry, business, market, demographics, geographic location, these four pillars will support your own culture where your people will be proud to go to work every day. Install these four pillars, and you will have the foundational support you need to create the business that all your stakeholders will be proud to be associated with.

## Care

*Life is short. Do whatever you can to help people - not for status, but because the 95-year-old you will be proud if you did help people and disappointed if you didn't.*

Marshall Goldsmith

Imagine the company environment where you and your people go to work every day, as a tribe you make a contribution for something bigger than yourself, you learn something new, you feel safe, and you go home happy. That’s what a *caring organization* is.

As I’ve witnessed the concept of Care come alive at WD-40 Company, I see our commitment to creating that caring organization flip all the levers that make life inside a healthy community a good, rewarding, fulfilling place to be. The caring culture is an environment where people are given the latitude to apply the principles of basic human kindness, gratitude for all that we have, the pursuit of justice, trust, transparency in our relationships throughout the organization, and the safe experience of honest conversation (which we will explore further in the next pillar – which is Candor).

Let’s first explore what the *caring organization* is not: It’s not Friday night gatherings where we all sing *Kumbaya*. It’s not about making decisions that have the least amount of negative impact on individual self-interests in the short-term. The caring organization is not a hug, a flower, or a brownie to soothe hurt feelings. The caring organization is not about creating a narrative where the CEO feels good about having the reputation for being a caring leader, at the expense of longer-term objectives.

There will be tough decisions. The caring organization strikes a balance between being tough-minded and tenderhearted. It’s unconditional love – to the extent appropriate in a business setting – combined with the commitment to doing what you need to do to keep people safe. The caring organization creates the environment that expects you to do what you must to execute on a rigorous business plan so that it endures over time. Caring also means that every tribe member has a responsibility to care about everybody else.

When I consider the fundamental elements of the caring organization, along with its most basic promise, the word trust comes to mind. It is the ultimate value proposition of the caring organization. To break this concept down in actionable components, we draw from Cynthia Olmstead's Trust Model, as described in *Trust Works!*, the book she wrote with Ken Blanchard and Martha Lawrence. It follows the simple ABCD format:

A – Are you *able*? Are you competent?

B – Are you *believable*? Do your actions reflect your words?

C – Are you *connected*? Do you take time to be with people in a meaningful, emotionally authentic way?

D – Are you *dependable*? Do you do what you say you're going to do? Can people rely on you?

Trust is the fundamental experience of the caring community. When you have trust, you have a tribe who will trust that you have their best interests always at heart. They will follow you into high-risk, long-term territory where great business outcomes will be found.

## Candor

*The void created by the failure to*
*communicate is soon filled with poison, drivel, and misrepresentation.*
C. Northcote Parkinson

The second element in Olmstead's Trust Model is B for believable. A culture where everyone is *believable* (not just the leaders) is one where everyone feels safe to speak their truth as they know it. This doesn't automatically presume that everyone will agree with each version of the stated truth. But without a culture of safety where everyone can be counted on to express themselves, the entire community will forfeit the enriching benefit of all points of view.

And, as Parkinson famously noted, the result of restrained truth is a toxic stew of half-truths, misunderstandings, critical decisions made based on only partial information, fractured relationships, stifled passions, and, before long, the regrettable departure of your most cherished talent.

This pillar of Candor shows up in actual behaviors: No lying, no faking, no hiding. Period. How this shows up to the leaders: They must always be prepared to be open and receptive to unpleasant information. How this shows up to the tribe as a whole: Each tribe member feels safe to take the risk of speaking the truth. In fact, when this pillar is installed correctly, on a cultural level, each tribe member feels more at risk for not speaking up.

The safety is in the communication. This is the open avenue to delivering the best self that every tribe member has. Truth, told respectfully and with positive intent, creates safety.

Most people don't consider themselves to be liars. But it's safe to say that many people will fake and/or hide when they feel they must protect their best self-interests. They fear reprisal.

Faking is simply not being true to yourself and your values. We've all heard, "Fake it until you make it." People think, "I'm just going to fake this because I'm afraid that people will see that I don't know something everyone else seems to know [for the record, they could be faking as well]." Or, "People will think less of me when they realize that I'm the only one in the room who has a completely different perspective on this matter."

Hiding occurs when there is something they don't disclose because there is fear of failure; fear of some sort of negative reaction, either from the tribe or tribe leaders; or fear of being caught in having done the wrong thing – or the right thing wrong.

In the psychologically safe workplace, we all hold dear the principle that when we behave in good faith and with good intentions, there is nothing we could do that would cause us to hide. And the spirit of the learning moment, which is such a crucial component of the psychologically safe workplace, is being true to ourselves and sharing our error with our tribe members. When we are, we are bringing additional wisdom and knowledge to the group as a whole. If we hide our mistakes, we deprive our entire team of essential learning.

As we have discussed above, the essential value of the psychologically safe workplace is the promotion of flow by the absence of friction. A workplace culture devoid of trust is an experience filled with friction. Candor removes that lack of clarity, deletes the confusion, smooths the surfaces of engagement of the rough splintery texture of emotional sticking points. And candor sets the stage for clear exchange of ideas. This is the value that promotes a high-performing workplace culture.

This isn't to say that candor-based conversations are easy. Some can be tough to initiate, some even tougher to be on the receiving end. Candor is not permission to be brutal, in the name of being honest. Candor must be accompanied by caring. All tribe members – especially leaders – should take extra care to fill the emotional bank accounts of their colleagues with positivity and supportive relationship interactions. This way, when the time comes for a conversation that requires uncomfortable candor, the trust is already there. Even the toughest conversation will result in strengthened trust that will prevail over the momentary discomfort of disclosure and discussion.

(For additional insight on this topic, read my blog post, When Caring Collides With Candor.)

## Accountability

*Accountability is hard. Blame is easy. One builds trust. The other destroys it.*
Simon Sinek

It seems that in these modern times, *accountability* is seen as a negative thing – an occasion to punish someone should that person fall short of the agreed upon standard or goal. That person is on the hot seat, and now must *account* for their disappointing performance.

At WD-40 Company, we have a different relationship with the word *accountability*. We see it as a two-way street in which leaders and their direct reports equally hold ownership of the way we perform our duties and what outcomes our efforts lead to. For this reason, for example, we tell our leaders that their job is to make sure their direct reports have everything they need to succeed in their jobs (for more information, read my blog post, Don't Mark My Paper, Help Me Get an A). And all our tribe members hold ownership of making sure they have what they need to succeed and lead the company to its fulfilled objectives.

Accountability, as a pillar, is a mutual discipline. But it's not the occasion to be disciplined. The disciplined commitment to results itself is, in practice, a freedom of sorts. When the company is committed to promoting accountability in its tribe, the individuals who demonstrate accountability also hold permission to do whatever is necessary to meet that accountability.

WD-40 Company's Maniac Pledge is an example of this philosophy in action. Years ago, a direct report spent a great deal of time explaining to me why a goal was not accomplished within the agreed-upon time. The blame was placed on a lack of critical information required to take the necessary action. After hearing the tribe member out, I observed the simple truth out loud, "In the same amount of time it took you to explain to me why this commitment didn't happen, you could have acquired the information you needed to get the job done."

And at that moment the Maniac Pledge was born, named after Aussie golfer Greg Norman, who was known for his maniac spirit. It reads this way:

*I am responsible for taking action, asking questions, getting answers, and making decisions. I won't wait for someone to tell me. If I need to know, I'm responsible for asking. I have no right to be offended that I didn't "get this sooner." If I'm doing something others should know about, I'm responsible for telling them.*

This spells freedom – the freedom that our tribe members feel is necessary to get their job done and meet their obligations to their own direct reports, as well as to their direct supervisors. In our tribe, we don't have the victim's attitude; there are many reasons and no excuses. We face the facts, learn and move to improve.

To us at WD-40 Company, accountability is manifested by the commitment that each tribe member holds in carrying through with their commitments. It is an understanding and expectation culturally that each one of us will own the desired outcome and all the steps required to achieve that outcome. For both ourselves and for the people throughout the organization chart who depend on us to help them be successful as well.

Accountability is achievement. But it's also learning that we are responsible for sharing with the rest of the tribe. And the community celebration when that outcome is realized.

It's fashionable to talk about "accountability partners" today. These are people we meet at the gym, on volunteer teams or in work groups who we must rely upon on regularly for mutual support in accomplishing our goals. But, really, our first accountability partner is the face in the mirror. Are we, as individuals, at peace with our actions and choices? If we have a face-to-face conversation (with our own faces) about how well we honor integrity to ourselves, how will we hold up in the investigation? As we say at WD-40 Company, when things go right, look out the window to find all the other people who contributed. When something goes awry, look in the mirror first.

We are each our own supervisors. And we are each our own direct reports. Are we giving ourselves what we need to make sure we succeed and that we help our company succeed?

### Responsibility

*Too many leaders act as if the sheep … their people … are there for the benefit of the shepherd, not that the shepherd has responsibility for the sheep.*

Ken Blanchard

Let's revisit the Maniac Pledge one more time. You probably have noticed that each item of the Pledge is driven by the words, "I am responsible for…." Responsible appears three times in the Maniac Pledge. In Accountability we speak of the outcome. In Responsibility, we speak of the relationship each tribe member has to the ideal that generates the outcome.

Simply put, *Responsibility* is a turbo-charged version of Accountability. As a tribe, now that we know what we're accountable for, it's our responsibility to make sure those desirable outcomes are actually realized. We each take personal ownership of the outcomes, and it's up to each of us to do whatever it takes to make it happen. Consequences (rewards and negative feedback in its variety of forms) are attached to the Responsibility pillar.

*Responsibility* is each tribe member's personal *relationship* to their role in realizing the ideal outcome. It demands that each tribe member respond with the answer "me," when the world poses the question, "Who is there to act?"

When it comes to psychological safety inside the workplace, when each tribe member is confident that everyone else shares responsibility in the company's success, the entire community feels safe to invest their hearts, minds, talents, efforts, risks in realizing the company vision that everyone agreed to. Everyone has their part and role. And everyone performs exactly as expected, because everyone shares the responsibility of a successful outcome.

I'm reminded of a strategy in rugby called the *blind pass*. In American football, even non-fans know how marvelous it is to watch as a quarterback throws a long pass into seemingly empty air – but with every expectation that a team member is on his way and will be positioned in the right place to receive the ball when it ends its flight and drops into his hands. You don't have to be an expert football fan to be impressed by the skill, strength, accuracy and teamwork of that play.

Well, in Australian rugby, that pass is basically performed backward – hence the name *blind pass*. The quarterback is running, but throws the ball backward, without being able to see whether there is going to be a team member likely to receive the ball. That is responsibility in action because the quarterback is psychologically safe in trusting that the pass will be completed and the objective of the play will be accomplished. There will be no wasted time, motion, effort, faith, trust, or ambition because everyone holds the assumption that the ideal of responsibility is equally shared amongst them all.

## Now You Have a Foundation

These four pillars are your foundation, upon which you can build strong and enduringly positive relationships among people who rely on your leadership. If you are the CEO, you have the highest likelihood of being able to spread this cultural foundation throughout the organization. If you are not the CEO, you can still influence other leaders by your example. People will want to work for you. When you have openings, internal candidates will flock to apply. Your group will meet and exceed objectives more often. Your employees will be coveted by other departments. You'll have opportunities for personal growth daily, because you will be investing in your own growth by working diligently to build and preserve the pillars of a fearless tribe!

***

To read this article online with embedded hypertext links, go here:

**https://www.linkedin.com/pulse/where-theres-friction-flow-four-pillars-fearless-tribe-garry-ridge/**

# About Garry O. Ridge

Chief Executive Officer and Director

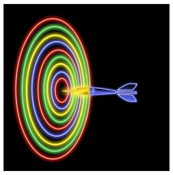
**Garry** O. Ridge is CEO and a member of the board of directors of WD-40 Company. He joined WD-40 Company in 1987 and held various leadership positions in the company before being appointed to CEO in 1997. He is also an adjunct professor at the University of San Diego where he teaches leadership development, talent management, and succession planning in the Master of Science in Executive Leadership program. He is passionate about the learning and empowering organizational culture he has helped establish at the WD-40 Company. In 2009, he co-authored a book with Ken Blanchard outlining his effective leadership techniques, titled “*Helping People Win at Work: A Business Philosophy called ‘Don’t Mark My Paper, Help Me Get an A.*’ ” A native of Australia, Mr. Ridge holds a certificate in Modern Retailing and wholesale distribution and a Master of Science in Executive Leadership from the University of San Diego.

For more information, visit his website at: https://thelearningmoment.net/

# “Agile Coaching” Is Not the Goal

By Johanna Rothman

Posted on December 18, 2018



I've met a number of agile coaches recently. They tell me they're hired as Scrum coaches or as Scrum Masters. They see their job as “better Scrum.”

It would be lovely if that was their one and only job. However, many of these coaches work in organizations just starting a cultural transformation.

Even though the client asked for agile coaching, that might not be what the client needs.

Instead of assuming you need “Better Agile” or “Better Scrum,” consider these questions:

- What business outcomes do you want to see, in 30, 60, 90 days? (Why the short-term thinking? Because consultants and coaches and agile approaches need to prove they offer value and find some quick wins.)
- What measurements will you require? (More on this below.)
- What are the boundaries of my role here?

These questions have nothing to do with a “better agile” or “better Scrum.” They are about business results.

Coaches—agile or otherwise—are not “installers” of a specific framework. If they are, they're teachers or something else. They're not coaching.

Coaches offer options with support. That means the coach often helps people think through alternatives (and maybe implement) alternatives. I don't see how to help think through alternatives without understanding the business results the organization needs.

Coaches who understand the organization's needs can and help the organization (or person or the team) achieve those needs.

That means that the very first job of a coach is to understand the metrics the managers want. What if the managers are “wrong” with their desire for specific metrics? The managers want something. Learn what that something is.

Here are things I often see managers ask for:

- Percent done or earned value. Managers want to know when they can capitalize or when the first release is ready. How can the coach help the team release more frequently? What are the impediments to frequent releasing? How can you visualize that data?
- “Accurate” estimation. Managers want to set expectations for people (inside and outside the organization). Maybe the coach can help create smaller stories or better roadmaps. Or, maybe the coach can show how the multitasking creates havoc with estimation and the project portfolio.
- How to see the value of a person. HR pressures managers for traditional “performance management.” How can a coach help a manager overcome that pressure and how can team members evaluate themselves or work with managers for more rapid feedback?

It's time to change the idea that the agile coach needs to help facilitate the team's working agreements or facilitate retrospectives first.

Yes, agile coaches might need to do that. However, I don't see how to create or facilitate successful agile teams without understanding what the goals are, how managers will measure success, and how to define a coach's success.

How can you offer options with support if you don't know where the organization wants to go?

That data goes to the organization's needs and how the organization will define the metrics around those needs.

Yes, that's coaching. It's not agile coaching. It's coaching. Agile coaching is a further refinement once you understand what the manager and org want.

Too often, I see agile coaches who don't know about lean thinking. They don't understand why managers might need to see cycle time, not velocity. They can only get to “Agile”-in-name-only, or other nonsense like that. Which might be much worse than not-agile-at-all.

If you're a coach, how can you work to support the people or the managers in their goals? How can you help them see alternatives? How will your services help them achieve business goals? Answer those questions.

If you work in an organization trying to use agile approaches, do consider joining us in the Influential Agile Leader workshop.

***

To read this article online with embedded hypertext links, go here:

**https://www.jrothman.com/mpd/2018/12/agile-coaching-is-not-the-goal/**

# About Johanna Rothman

**Johanna** Rothman, known as the “Pragmatic Manager,” provides frank advice for your tough problems. She helps leaders and teams see problems, resolve risks, and manage their product development.

Johanna was the Agile 2009 conference chair and was the co-chair of the first edition of the Agile Practice Guide. Johanna is the author of 14 books that range from hiring, to project management, program management, project portfolio management, and management. Her most recent books are *From Chaos to Successful Distributed Agile Teams* (with Mark Kilby) and *Create Your Successful Agile Project: Collaborate, Measure, Estimate, Deliver*.

Read her blogs, email newsletter, and more information about her books at www.jrothman.com.

# A Smashing Retrospective!

By JM Roxas

Posted on March 14, 2018

Retrospectives are one of the key activities that teams have in order to improve. As a facilitator its always exciting when you get to try new formats or ways of conducting one. The board game King of Tokyo is one of my favorite board games and I thought it would be fun if we could somehow use the mechanics of the game to conduct retrospectives. Here is how we did it.

### What you need to get started

1. King of Tokyo board game. The Power Up! expansion is not necessary but is nice to have.
2. Space to conduct your retrospective where there is a table or surface in the middle of where the participants will be sitting.
3. Your team

## How to run the retrospective

The next sections will describe the different parts of the activity and how we did it.

### Prepare the space for the retrospective

1. Place the monster cards face up on the table. Make sure all of them are visible.
2. Have the team sit around the table or surface.

### Select your monster!

The select your monster activity serves as a check-in for the team and warm-up for the main event. We start by having each member of the team select a monster that will be their character for the game. The instruction is to select a monster that you feel represents you or how you feel about the last iteration. There are many ways that a participant can relate to the monster they choose. For those familiar with the game they might select monsters based on their skills and how those skills are similar to their personality. For others, they simply choose them based on aesthetics or because its cute.

After everyone has selected their monster they present which one they selected to the group then explains why that monster was chosen. The last thing to do would be to give out the monster tokens for the participants, remove the monster cards, and place the game board at the middle of the table. Once that’s done its time to move on to the next activity.

## Time to play!

Explaining the rules of the game is the next step which is done right after everyone is done with the select your monster activity. The rules we used are as follows

1. Each player gets one turn per round and the game will last a minimum of 3 rounds. After the 3rd round, the team will decide whether to keep playing or stop.
2. At the start of a player’s turn, he or she will roll one dice. The result of the dice will determine the action that the player will take. The different actions are different types of questions or feedback that the player has to give. These actions are described below.
3. The first player who rolls the claw will automatically enter Tokyo. After another player has rolled a claw the player inside Tokyo can then choose to leave and the player who rolled the claw will replace him inside Tokyo.
4. All players who are outside of Tokyo will be giving feedback or answering a question related to the person inside of Tokyo. While the person inside of Tokyo will be giving feedback or answering questions with regards to the whole team.
5. While the game is going on, someone from the team or the one facilitating the retrospective should note down the claw and energy feedback being given by the team. These will be used in the latter part of the activity.

## The different actions

The dice has six faces but only three actions are used. The 1,2,3 numbers on the dice would be partnered with either the heart, claw or energy symbol. Here are the actions we did based on the dice:

### Heart or #1

Heart in the game stands for recovering health. In our version heart is positive feedback you’d like to give the person or the team. Some examples of heart feedback would be: “Hey John, I really appreciate you helping me figure out that bug yesterday” or “I really love that as a team we play music at our desks and we get to bond over songs we like”.

### Claw or #2

Claw in the game is the SMASH! action. In our version claw is our way of pointing out habits we should SMASH! or things we did or said that made us want to smash things. In short rolling, a claw would mean the player would have to give constructive feedback or something that can be improved on. Some examples of these would be: “Donald, you really need to run those unit tests before committing code. You broke the build three times last week” or “As a team, we all need to improve on writing things down. We always forget what we agreed upon!”

### Energy or #3

Energy in the original game would be used to gather energy bricks which would allow you to buy cards that give you special skills. For our retrospective energy meant the player would have to propose an experiment to energize the team. Some examples of this are: “I think we should make a dashboard where we can check all our test environments” or “I’d really like for us to try having our stand up in the afternoon instead of first thing in the morning”.

## Action items time!

After three rounds or when the team has decided to no longer play one more round its time to start the last part of the activity. The last part is where the team goes over their claw and energy feedback and decides what they feel most strongly about and discuss action items they'd like to do to address these.

---

## Some variations to the format

### "Open space" time before making action items

After the team has decided to no longer play another round we typically do an open space type of activity. We dedicate a small amount of time where the team can just freely exchange feedback with each other. This is to give everyone a chance to share something that they might have not gotten to because of the dice rolls.

### Changing seats after each round

This is one way to get more combinations of people giving feedback to one another. For example, if Tom takes his turn after Gina then it would be unlikely that Tom gets a chance to take an action directed at Gina. Shuffling or changing seats after one round helps mitigate this.

### Loading the dice

Since the numbers 1,2,3 are partnered with heart, claw or energy then we can easily load the dice to get more targeted feedback. For example, a team that is going through a rough time might want to agree that 1 and 2 are heart, while 3 is claw. This increases the chances of them giving positive feedback towards each other and lifting their morale. The flip side of that would be a team that is doing very well might want to load up on claw and energy so they can get more ideas on how to push themselves further.

### Can't stay too long!

In one smaller group where we've done this, we've agreed that if a player completes one round without getting kicked out of Tokyo then that player gets kicked out. This was to avoid one person getting all the attention during the retrospective and as a way to balance the game in the chance that no one rolls a claw for one round.

### Re-roll the dice

We've tried allowing each player one re-roll for the whole game. This added a layer of strategy for some, where they'd re-roll to avoid giving feedback their uncomfortable with. For others, it added some comedic value when someone re-rolled and got the same result anyway. This worked best when they didn't know we would have some open-space time in the end.

## Some key learning we've had

1. The uncertainty of the dice gives a sense of excitement during the game. This keeps people interested in what the next roll of the dice is. This also extends to people being excited about their turn. So keeping track of how long it is before it's their turn again is something that's important.
2. Having the dice tell us what kind of feedback to give makes us step out of our comfort zone and give feedback we might not normally give. An example of this would if you had certain members of the team that don't talk often, they're also not likely to give feedback towards each other. This gives them/makes them do that.
3. The energy/experiment feedback is always the one people have the most problems with.
4. The inside Tokyo gameplay mechanic allows people to be more personal with feedback. This is something to watch out for while facilitating the retrospective as some people might not best at giving claw feedback towards other people. Thankfully I've never had a problem with this.

## Last thoughts

After doing this with several teams I'm inclined to use it again in the future. I've found that just the image of someone walking into the retrospective holding the King of Tokyo box is enough to get most people interested so that's always a plus when doing retrospectives. As with all formats, this one isn't something I'd use every time as the novelty would wear off but it is one that I feel would energize teams or at the very least pique their interest.

***

To read this article online with embedded hypertext links, go here:

**https://www.jmroxas.com/2018/03/15/a-smashing-retrospective/***

(*The article was published in Hackernoon at the time it was nominated for this publication.)

# About JM Roxas

**JM** Roxas is an agile coach from the Philippines with years of experience working in the software industry. From his early days a developer he's had a passion for making the workplace fun and exciting. As an agile coach he has coached teams to bring their whole selves to work and to tweak their agile practices towards their own tendencies. JM is also the author of The Agilist Field Guide a book that helps newcomers start their agile journey. More of his writing can be found at his personal website jm-roxas.com

# How To Change Your Organizational Culture

By Michael Sahota

Posted on July 12, 2018

You will learn How to Change Your Organizational Culture. Yes. How to Change Your Culture. It requires effort and focus. And it is possible. I have done it and leaders around the globe have applied this same information to change their culture. What follows below is an outline of the proven steps. I also include pointers to supporting resources.

## 1. Desire for Growth

The starting place for culture change is desire – a powerful urge to create change. Nothing less than this will result in success. So that is the starting place. Anyone interested in shifting culture needs to look inside to see what is driving them and make sure the have motivation to do the work needed.

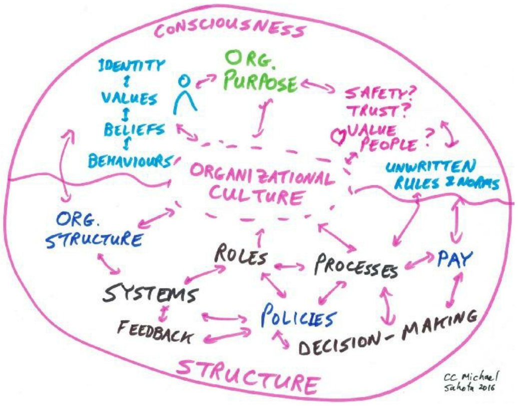
I used to subscribe to Kotter's "Sense of Urgency" and even advocate this. I no longer do. It turns out that Urgency linked to fear and a lower level of psychological safety. This inhibits personal and organizational growth. For this key reason, "Desire" is a better choice.

**Strong Desire for Growth is Essential**

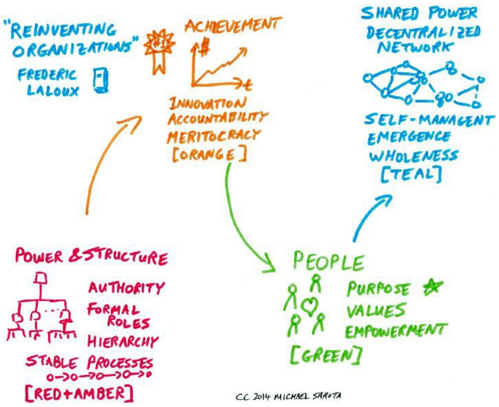
Organizations that have sustained organizational growth over decades see improving as part of everyday work. They invest in growth because it is important. It's the right thing to do. Not because of urgency.

## 2. Understand Existing Culture

The next step is to understand your existing culture. But what is Culture? We can define it as "How we do things around here". I have experimented with a lot of culture models and recommend the two that are proven in terms of simplicity and power. You can use them together to diagnose and your culture and orient for growth.



The Sahota Culture Model provides a clear understanding of culture through identification of the interconnected elements that shape culture. It also highlights the need to focus not just on Structures but also on the Consciousness (or Mindset) of a system. We often fall into the trap of focusing on structures (especially process) rather than focus on the people and how they are working together. This model reminds us that it's really about the consciousness (or mindset), the people, not about the structures or process.



The other model that is very powerful is a modified version of the Laloux Culture Model. It may be used to assess where the organization is right now. It also has the tendency to help spark a desire for shifting to a higher performance way of working such as Green or Teal. One key reason to use this model is that it has heaps of case studies and research to support the claims of high performance. It also lines up with many other models and theories of culture and behaviour such as McGregor's Theory X – Theory Y.

## 3. Create a Star on the Horizon

The next step is to look at case studies and examples of the kind of company that you want become. There are lot's of great resources such as the book "Reinventing Organizations" or Diverse Paths to High-Performance Organizational Culture.

It is a good idea to use these for inspiration. The goal is to create a "Star on the Horizon" that is aligned with the desire for change. Don't try and copy structures. Copying simply gives you the structures without the shift in culture.

> ***Find Your Own Path.***

The secret here is to find your own path. Selecting a path is primarily a function of two things: 1. The existing situation in your organization. We can only grow and evolve from the place we are at. 2. The shared desire of people to create a new future. The desire could just be top leadership or they may co-create this with people throughout the organization.

## 4. Culture Grows Locally

A common misconception is that culture change is for the whole organization. It is important to understand that in most organizations culture varies by team, department, and location. It is as unique as each individual manager. So keep in mind this key point:

> ***Culture is a local phenomenon.***

Since it is a local phenomenon, it means that it is possible to make changes locally within your part of the organization. The most common way for culture to grow is Culture Bubbles. Of course when we do this, there will be culture gaps that create tension and challenges.

The key idea for reducing the tension is to Build Culture Adapters. There will be different ways of working and different values inside the bubble and outside the bubble. The adapter idea is to reduce conflict with the rest of the organization by building adapters between the ways of working. It's a key pattern for creating sustainable culture bubbles.

## 5. Leaders Go First

Culture is primarily a reflection of Leadership. What happens at the bottom of the organization is a fractal of what happens at the top of the organization. (Thanks to Glenda Eoyang for this wisdom). It is well known that the performance of a team is a direct reflection of their manager – this was proven through validated real-world research almost 20 years ago through the Gallup 12 “Engagement” Questions.

> ***Culture Change is Not Delegatable***

The way to change culture is for leaders to change how they interact with people and the organizational system. A key concept here is that Organizational Behaviour Follows Leadership Behaviour. A new kind of organizational behaviour way of working requires that leaders behave in a new way of working. So successful transformation requires that Leaders Go First!

## 6. Leadership Growth is Required

A key lesson in my career is that the Leader is the Limit for Growth. I notice that to create high performance organizational systems, leaders needed to develop themselves as human beings. They needed to grow into the kind of leaders we see in high performance environments. This means inner work cultivating trust, safety, and connection. As leaders we need to get our egos under control so we can develop leaders around us.

This is not for the faint of heart. We are talking about developing ourselves not just as leaders but as human beings. Like you, I am on this journey too. I created the 4A’s Conscious Leadership Model to capture the step-by-step approach I have been using to grow myself. It’s a powerful tool to help rewire our unconscious behaviours that are preventing us from showing up as the leader we desire to be. We are so deeply conditioned by society to have behaviours that are contradictory to high performance. Dedicated focus and effort is required to shift our habits and unconscious behaviours.

> ***A learning organization is a place where everyone grows.***

Remember the desire for organizational growth in step 1? This is where you need it. Personal growth requires a strong drive to keep up the effort.

This is the secret of changing culture: All we need to do is change our behaviour and culture will follow.

## It’s a Journey

The above steps are sufficient and necessary for culture change in an organization. What is shared here is the key starting elements for culture change. Or course there are a lot more details on how to do the steps outlined here and even more on supporting the journey.

## You Can Do This Regardless of Your Role

Execs, managers and coaches that I have trained have successfully applied what I am sharing here. We are all leaders. We may be a leader because people report to us or we have more seniority or expertise. And we can also be a leader because of how we choose to show up.

## You Can Implement This Immediately

Regardless of your role you can chose to show up NOW the way a leader of the future organization would. You have full control over your behaviour.

> **You don’t need permission, budget or authority.**

You don’t need permission, budget or authority to start acting in ways that model high performance behaviour. All of us can shift our local culture immediately. The only limit here is your desire and your investment in developing yourself.

It’s a big shift for us as leaders. Sure we still need to support the development of people around us so that we have leaders at all levels. But this is secondary to growing ourselves to fully model the kind of organizational leader needed for the future organizational culture/organization we wish to create.

## Summary

So here are the six key steps to change your culture:

1. Desire for Growth
2. Understand Existing Culture
3. Create a Star on the Horizon
4. Grow Culture Locally
5. Leaders Go First
6. Leadership Growth is Required

And here are the important tips to keep in mind:

- It’s a Journey
- You Can Do This Regardless of Your Role
- You Can Implement this Immediately
- You do not need permission, budget or authority

Michael K Sahota guides and teaches leaders how to create high-performance organizations. He is the creator of a proven system for leading change through a practical playbook. His model for Consciously Approaching Agile guides the creation of a cultural and leadership context where Agile creates lasting organizational results. Michael has taught over 1000 leaders worldwide through his highly acclaimed “Agile” Culture & Leadership Training. His company, Agilitrix, delivers “Agile Transformation- Redefined” with an elite team of consultants who specialize in unlocking Culture.

***

To read this article online with embedded hypertext links, go here:

**http://agilitrix.com/2018/07/change-organizational-culture/**

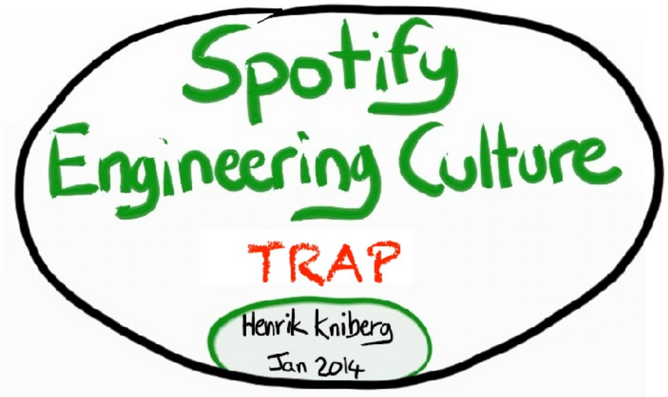
# How To Replicate Spotify’s Success

By Michael Sahota

Posted on November 22, 2018

In this post, you will learn the how to replicated Spotify’s success. How to create a high performance organization by copying “The Spotify Approach”.

They actually mention the key pattern in the very famous video created by Henrik Kniberg. But everyone seems to fall into the usual traps and are unable to replicate Spotify.



## The Trap: Copying Structures

Let’s start with the trap.

There is a lot of excitement and interest in Spotify as a case study for high performance with Agile. At trainings and at meetups all over the world, I ask about people’s efforts to replicate Spotify and there is an almost a universal response: that people are not successful. Here is what usually happens:

1. People copy Spotify’s structures such as Squads, Guilds, & Chapters.
2. It doesn’t deliver the expected benefits.

What’s going on?

*Copying other people’s structures does not work.*

It is a very good idea to learn from others. To get inspiration. To get ideas. But not to copy.

## The “Spotify Approach”

If you listen carefully to the Spotify video, they explain clearly what they did to create success. What you can do to create success. Here is what to do:

Look at where you are and grow from there. *Don’t try to copy anyone else.*

The *whole reason* they created the names “Squads”, “Guilds”, and “Chapters” is so they could make up a solution that was right for them. So they wouldn’t fall into the trap of copying someone else’s solution.  That’s a key pattern for success: Create a solution based on your situation.

*Find your own path.*

## How to Replicate Spotify's Success

OK. But how do we replicate Spotify's Success?

When listening to the video, it is clear that they had an organizational culture that supports trust, innovation, autonomy, experimentation and little bureaucracy. And with these cultural elements in place, they could evolve a very powerful solution to unlock their potential. And keep evolving it. They continue on their path today. For example, they are baffled that anyone would copy the "Spotify Model" since they themselves don't follow it.

Here is what you do:

> *Create an organizational culture that supports trust, innovation, autonomy, experimentation with little bureaucracy.*

Or more realistically move your organizational culture in this direction. If you are normal, your culture will need a fair bit of work to approach Spotify's.

That is how to replicate Spotify.

*Emulate Spotify's Culture, not it's Structures.*

Just to clarify, I am not literally suggesting copying Spotify's Culture. What I advise is to look at their culture and other high performance organizations for inspiration. Then find your own path based on your organization's history and current situation.

## 3 Key Take-Aways

Here are the three key take-aways from this post:

1. Don't fall into the trap of copying other people structures
2. Find your own path; your own solutions.
3. Use Spotify and other exemplary organizations to inspire your own organizational culture initiative.

## Thank You: Henrik Kniberg& Spotify

First, a personal thank you to Henrik Kniberg for all the amazing work he has done in the community and how he has personally helped me through his writing, videos, talks and personal communication. In particular the video created on Spotify has inspired a lot of change worldwide.

Second, a big thank you to Spotify for publicly sharing about it's Engineering Culture. They are a great case study for possible elements of high performance organizations.

## Where to Learn More

Culture change isn't easy and there is a lot to say on this. If you want to learn more here are some additional resources:

- Consciously Approaching Agile for Lasting High Performance
- How to Change Your Culture
- How To Be Successful with Agile in Any Culture

If you want the detailed “HowTo” on shifting your leadership and culture to high performance we suggest you consider our in-person or online training or advisory services.

Michael K Sahota guides and teaches leaders how to create high-performance organizations. He is the creator of a proven system for leading change through a practical playbook. His model for Consciously Approaching Agile guides the creation of a cultural and leadership context where Agile creates lasting organizational results. Michael has taught over 1000 leaders worldwide through his highly acclaimed “Agile” Culture & Leadership Training. His company, Agilitrix, delivers “Agile Transformation- Redefined” with an elite team of consultants who specialize in unlocking Culture.

***

To read this article online with embedded hypertext links, go here:

**http://agilitrix.com/2018/11/how-to-replicate-spotify-success/**

# How to Overcome Resistance in Your Agile Transformation – The Power of Invitation

By Michael Sahota

Posted on December 3, 2018

Almost all Agile Transformation program experience resistance from managers, workers, other departments, etc. In this post, we will explain why you are experiencing resistance and how to eliminate or reduce it.

Why are we experiencing resistance to Agile? Isn't it about making people's work environments better so they can be productive? Let's assume for a moment that Agile is a good idea. Then what is going wrong with how we introduce it into organizations?

## Push creates Resistance

The key challenge with "transformation as usual" is that change is pushed onto people without their involvement in the decision. And they resist that.

## How Does Push Impact You?

There are many ways to create resistance that are variations of push. As you read the words below, think about how it feels for you when someone in authority behaves that way towards you:

- Tell
- Sell
- Make
- Convince

For most people, these words will not feel comfortable.

We don't like it when people tell us what to do. We don't like it when someone makes us do something. We don't like it when people try to sell us something or convince us of their point of view. When these things happen to us, we resist it.

## Agile Transformation Anti-Patterns

It turns out everyone else has the same response that we do. The more we push, the more resistance that we create.

Here are some key anti-patterns to help you understand how we author our own failed Agile Transformations by creating resistance:

- **Mandated Whole Organization/Department Change.** Lot's of people see this as a great success: "Hurray, everyone has to do Agile now". But the victory is hollow and short-lived as this soon turns into a widespread case of "Cargo Cult Agile" where many people do the practices but nothing has really changed except the names.
- **Agile Evangelism.** I still encounter a lot of people in our Leadership trainings that see themselves as Agile Evangelists. The positive aspect is the desire to make things better. The negative or destructive aspect is to sell and convince. This creates resistance and damage that usually outweighs the positive. Every successful change

program I have seen was about more than Agile. See: Why Workshop for more information.
- **Agile Metrics (The Agile Police).** Another well-intentioned approach that goes horribly wrong in practice to measure how “Agile” teams are and set targets for them to become “More Agile”. What this tells people very clearly is that Doing Agile is more important than Being Agile. You have to do Agile if you want to succeed around here or get your bonus. What this signals is that the process is more important than the people – the exact opposite of Agile.

## Agile is about Pull

When you go to the core of what Agile is really about – it is based not on Push but on Pull. Like Lean, Scrum teams pull work they can complete in the next Sprint. Kanban teams pull tickets when they are ready.

### Push Cannot Create Pull

How can we use a push approach to create an environment that supports Pull? It doesn’t make any sense. The evidence shows that using push simply doesn’t work. Sure, we might be able to create something that on the surface looks like it’s working. However, something different is needed to actually get the deep shift in behaviour the motivated, engaged people that is needed for pull to work.

## What is Pull?

Similar to the list we had for push, we can create the opposite list for pull.

As you read through the words below, think how they feel:

- Invite
- Listen
- Inspire
- Co-create

For most people, these words will feel positive. That’s the effect we want to create in organizations to foster success.

We like it when we have freedom to choose. We like it when people invite us and it is optional for us. So we can decide for ourselves. We like it when people listen to us about what we want. It feels really good when we are inspired and excited about something. We like it when we get to be involved in decisions that impact us.

Can you imagine how effective your change will program will be when people feel this way?

## Pull Success Patterns

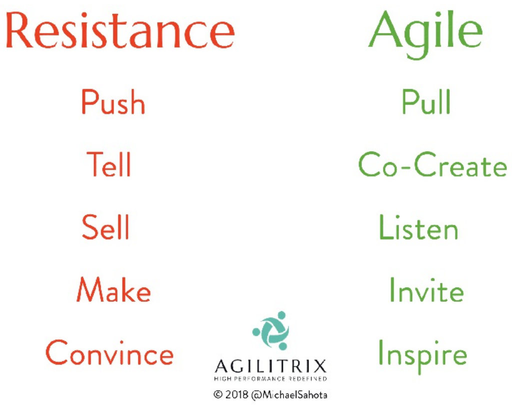
It is much easier to see what we are doing wrong than to look at what we are doing right. Let’s look at some key success patterns using pull. If you are not familiar with these, or are skeptical, we invite you to run an experiment to try them out.

- **Listen**. Most people want to be successful. Most teams want to be successful. One big challenge is that we often don’t take the time to listen to people about what they want and think. Take the time to listen to people about what they want. And help them with they want. List to them first, before telling them what you think or want. You will get a lot further.
- **Go Where the Energy Is.** The way to create a widespread success is to make small successes and build on them. Work with the people or teams that actually want help right now. Give them the help they want (listen) and make them successful. After time other people or teams may be ready or they may see what is happening and want it too.
- **Co-Create Solutions**. Agile is about collaboration, right? Isn’t it obvious if we are following an Agile approach we will collaborate with people on the transformation itself? I like the term co-create since it is about

having people impacted by change be involved with the change. This is similar to the approach advocated by Lean Change and Open Space Agility. One difference in what I have seen work is that notice that managers and executives also need to be involved in this process alongside workers. Of course, that usually entails training for managers so they learn how to give away power in ways that help and do not harm.

## Push vs. Pull

The following image is helpful for comparing and contrasting the choices we have when working with people and teams on our Agile Transformation programs.



| Resistance | Agile |
|---|---|
| Push | Pull |
| Tell | Co-Create |
| Sell | Listen |
| Make | Invite |
| Convince | Inspire |

AGILITRIX HIGH PERFORMANCE REDEFINED

© 2018 @MichaelSahota

A key challenge to shifting our mindset to this new way of working (Agile) is that it feels uncomfortable. In particular, we often have a low skill level with the pull words. Really take time to see how you may inadvertently create resistance using the push words. Once we understand that the push words really do not work, it may give us the courage to be a novice operating using a pull model. And there is a lot more to say about operating in a people-friendly (pull) way, so if you think this isn't the whole story, you are correct.

## Agile is about People

The core of Agile is about "People over Process". The key success pattern at the core of Agile is that success comes when we pay more attention to people than process. Most Agile Transformation get this backwards and focus on process. The usual push behaviours are a clear demonstration that most "Agile Transformations" are not really about Agile at all.

## Wave2Agile

We need a fundamental rethink of Agile to overcome todays challenges. We are calling this reinvention Wave2 of Agile. It means putting people first. Inviting change. Dropping the destructive push behaviours. It isn't about making anyone do this. It's about us a leaders starting with ourselves to model the behaviours we want to see in others. Are you ready to join Wave2Agile?

## What to do Tomorrow

1. Notice where other people are using push behaviour and how it impacts you.
2. Notice where you are using push behaviour and how it is creating resistance.
3. Increase your "Agile/Pull" activities and see how people respond.

## Where to learn more

I have a free online training that explores push vs pull and related topics for using an Agile Mindset to create success: “Top 3 Secrets of Agile Transformation: Using the Agile Mindset”.

***

To read this article online with embedded hypertext links, go here:

**http://agilitrix.com/2018/12/how-to-overcome-resistance-in-your-agile-transformation-the-power-of-invitation/**

# About Michael Sahota

**Michael K Sahota** trains and guides leaders how to create high-performance organizations. His highly accoladed *“Agile” Culture & Leadership (CAL1) Training* gives gives managers and coaches worldwide the mindset shift needed for lasting success. In 2012, he published the ground-breaking book "*An Agile Adoption and Transformation Survival Guide: Working with Organizational Culture*".

# 5 Ways ScrumMasters Can Enhance Daily Standups

By Ajeet Singh

Posted on October 31, 2018

## Summary:

Daily standup meetings can turn into a perfunctory chore, with everyone simply going through the motions. It’s the ScrumMaster’s job to make sure that doesn’t happen and the meetings remain useful for everyone. With these five ideas, the ScrumMaster can actively help daily scrums be effective and encourage communication, transparency, and efficient delivery of value.

Daily scrum meetings are primarily for development teams to sync up, discuss progress, and plan their work for the next twenty-four-hour period. A ScrumMaster typically serves as a facilitator for daily scrums, and other stakeholders can listen in too.

Teams are free to decide how they wish to utilize these fifteen-minute meetings, but this brings additional onus on the ScrumMaster to ensure that the meeting still results in two critical outcomes:

- The team is focusing on the crucial aspects of progress, including how to remove impediments, and is able to put a realistic plan for the day in place
- The product owner and stakeholders are able to gain vital information on progress toward giving customers more value and producing releasable code

Here are five ways the ScrumMaster can better facilitate daily scrums to achieve these outcomes and encourage communication, transparency, and efficient delivery of value.

## 1. Tightly control work in progress

Scenario: A developer in the daily scrum meeting says she is planning to work on a new story and doesn’t see any immediate impediments. The ScrumMaster asks whether the story she was working on yesterday is finished, and the developer reveals that the story has been put on hold due to a technical architecture issue that an architect must look into, so she is planning to work on something else.

It’s a good thing the ScrumMaster asked questions to unearth that a story is on hold and there are impediments that have not been discussed or addressed. Now, the ScrumMaster may suggest that he and the developer meet immediately with an architect to expedite removal of the blocker so the highest-priority work can continue.

Doing so controls work in progress (WIP) by focusing on completing work that has been started before beginning something new. This also reminds the team to keep track of how many stories are being worked on at a time and keeping WIP under a certain limit agreed to by the team.

## 2. Encourage collaboration on daily activities

Scenario: A tester in the daily scrum meeting says he will continue testing the story picked up yesterday and there are no impediments. The ScrumMaster asks if this testing may be completed before the end of the day and, if so, what he will be doing next. The tester realizes he forgot to mention that he will likely pick up a new story to test sometime after lunch, but he hasn’t discussed this with the developer yet.

The ScrumMaster made sure that critical collaboration is happening when transitioning from one story to the next. The benefit is that because the entire team knows what is happening, anybody who has any knowledge about this new story can provide information.

It’s essential that the ScrumMaster makes sure development and testing are collaborating every day and discussing stories as they are being implemented and tested.

## 3. Identify all blockers

Scenario: The database administrator says she finished one story yesterday and picked up another that she plans to continue today, and that there are no impediments. But the ScrumMaster remembers that there were some recent email exchanges between this database administrator and the client SMEs seeking clarifications on certain points on this story, so the ScrumMaster asks whether these issues have been resolved. The database administrator acknowledges that there is more to clarify for this story, and there are actually blockers that the ScrumMaster or someone else can help with.

It’s important for the ScrumMaster to track all unresolved issues and make sure the team openly discusses them so anyone with the ability can help unblock them quickly.

The benefit that this facilitation brings is that the team becomes more attentive to outstanding issues and remembers to leverage the ScrumMaster for help in unblocking issues.

## 4. Help the team prioritize issues

Scenario: A team is working on production issues and new feature development, causing them to have to allocate their work in both of these areas. The ScrumMaster asks questions about the high-priority production issues: Are they being adequately prioritized? Are they evenly distributed throughout the team, or at least properly planned to accommodate everyone? And will they impact the completion of planned new features? The team realizes it won’t get all of the new features completed, so they discuss allocation with the product owner to get her guidance.

The ScrumMaster should help the team continually set expectations with the business side and better negotiate scope for development in upcoming sprints. This way, the business side always knows where things stand and is never surprised at the end of a sprint.

## 5. Support alternative daily scrum formats

Scenario: Instead of using the standard daily scrum format of answering three questions—What did you do yesterday? What will you do today? What is blocking progress?—the team decides to discuss each story one at a time, with those directly involved speaking. Unfortunately, this causes meetings to take longer than expected.

During the next retrospective, the ScrumMaster suggests the team look at ways of keeping their daily scrum to fifteen minutes. The ScrumMaster also starts keeping an eye on the clock during daily scrums, helping the team learn how to get their updates communicated effectively.

Because this is the team’s meeting and they should self-organize to decide what they want, the ScrumMaster should support alternative daily scrum formats if that’s what the team thinks will make sense for them. But it’s still top priority that the ScrumMaster makes sure the goals of the standup are met as well.

Daily standup meetings can turn into a perfunctory chore, with everyone simply going through the motions. It’s the ScrumMaster’s job to make sure that doesn’t happen and the meetings remain useful for everyone. With these five ideas, the ScrumMaster can actively help daily scrums be effective and achieve their intended purpose.

***

To read this article online with embedded hypertext links, go here:

**https://www.agileconnection.com/article/5-ways-scrummasters-can-enhance-daily-standups**

# About Ajeet Singh

**Ajeet** Singh is an IT leader with 18 years of diversified experience in the areas of Application Development, System Integration and Software Testing. He has been working on Agile track for over 4 years, serving agile teams across geographies of USA, UK and Australia in varied roles of Agile Coach, Lead Scrum Master and Delivery Manager.

# The 2T Factor of Being Agile

By Sourav Singla

Posted on April 26, 2018

Introspect with all the below said questions & help yourself to know where YOU stand today in Agility?

- *Is your team's daily scrum becoming merely a status meeting?*
- *Is your team's Product Owner challenging development team's estimates during story grooming?*
- *Is your development team complaining on every other story that it needs further grooming though story is meeting the definition of ready?*
- *Is your team's product owner stating something like this "Why do you need X hrs? I think it is doable in an hour."*

*If YES, as a coach it is of grave concern to you since these are the leading indicators of failure. It may lead to "Gun Point Agile or Lipstick Agile" (Courtesy TCS Agile Ninja workshop).*

*Gun Point Agile: Team is bound to deliver through command & control with fixed time cost & scope.*

*Lipstick Agile: Actually teams are working in waterfall methodology with few of scrum practices incorporated and they call themselves as agile scrum team!*

***Can you sense the reason behind it??*** **I will call it as the 2T factor in Agile (Transparency & Trust).** It can be the Missing transparency between Product Owner & development team leading to lack of trust between them. This is also one of the most common challenges I encountered & one of the biggest roadblock to agile transformation.

Actually trust is like a savings account. In order to open the account, we need an initial deposit, similarly, when an organization decides to “adopt Agile,” leadership needs to invest in their teams, granting an initial deposit of trust by trusting the team until there is a reason **to NOT trust the team**.

Why Trust is important?

Let’s refer the 5th agile principle which talks about trust: *"Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done."* I also feel Trust is also the foundation for the 5 Scrum values, *Openness, Courage, Respect, Commitment and Focus*.

**Without trust these values cannot be achieved.** Moreover Trust contributes to the psychological safety of the team which impacts team effectiveness. If the team does not feel safe, **inspection**, **experimentation** and **reflection** is not possible, must for any team to be successful.

Common reasons on how does team break trust from Product Owner (PO)**:**

1) **Partially done work** If team often delivers partially done work by the end of the sprint, PO has difficulty to know where we are and loses the flexibility to adapt in next sprint, team breaks trust from PO.

2) **No Transparency** Team states that the work is done, while it is not. Later, PO finds it out. Team breaks trust from PO.

3) **Over Sizing** If team overemphasize the safety in delivering their commitment, it does not set challenging goal for themselves, team breaks trust from PO.

4) **Inconsistency in Velocity** As PO uses velocity for long-term planning, while velocity varies greatly, PO loses predictability. When team's velocity does not get stabilized after a while, team breaks trust from PO.

Common reasons on how Product Owner (PO) breaks trust from team**:**

1) **Command & Control** PO micro-manages the progress within the sprint through excel trackers.

2) **PO committing to stories rather than team** During sprint planning, PO pushes team to commit stories in a sprint.

3) **Deadlines and Forced commitments** Instead of targets and goals which can inspire the team, PO is busy in giving & tracking deadlines.

4) **PO expecting exponential increase in velocity** PO expecting increase in velocity in every sprint, eventually pushing the team to over commit.

5) **PO starts deciding the "HOW" part** If PO enters into the implementation domain and interferes team from self-organizing on how, he breaks trust from team. Let Team decides how.

6) **No Clarity on Project goal** If Product Owner has not shared the project vision/goals with the team, team did not feel connected with the project & PO.

**What an agile coach can do?**

A coach can take various stances.

**Step 1:** As a coach, I generally try to dig deeper into the real reasons of trust deficit between team & PO. For that I do *trust assessment exercise.*

**Trust Assessment Exercise**

A) Trust assessment exercise for Product Owner where questions can be asked like *"Team is not worthy of being trusted"* to extreme *"I fully believe all my team members"*. Similarly other questions can be like *"My team asks for my approval on everything"* to "My teams take risks to achieve the sprint goal". Scores are associated with each option like score is low where **there is more command & control** & vice versa.

B) Trust assessment exercise for Development Team which suggests how team members feel about the level of trust coming from Product Owner (PO). Here questions can be asked like "*I have to take approval before working on any story"* **to** *"I am given 100% freedom to do my work"*. Again questions which states there is command & control to empowerment. Scores are associated with each option like where there is more command & control, score is low & vice versa.

So based on the scores, **below three scenarios can arise:**

1) PO and team scores are identical with average score on higher side. It suggests everyone is on the same page & there is good bonding/trust between PO & team.

2) PO and team scores are identical but average score on lower side. It suggests everyone is on same page but there is lack of trust between PO & team.

3) There is a gap between PO & team scores. It suggests a lack of openness & trust deficit between PO & development team.

**Step 2:** If scores are on lower side or there is huge gap (Scenario 2 & 3) below things can be tried:

1. **Common Understanding of Trust** Run a brainstorming session using trust canvas where teams suggests what trust means to them & then showing that canvas to PO using below questions.

- Which things can add up the trust
- Which factors are important for creating trust
- What are the things one does not have control but are leading to distrust & lack of transparency

**Thus creating a common understanding of what trust means to them.**

2. **Pairing Up** Invite development team members to pair with PO for a day & have them sit in on a stakeholder PBI roadmap discussion. Let them get good insight into what a typical PO day is like & see the PO world. Team will be able to understand the challenges of PO.
3. **Management 3.0 Delegation Poker exercise** Facilitate a delegation poker excise. It allows Product owner to uncover disconnects in servant leadership. It provides a good platform to team members to talk openly with the PO & find answers to their own problems.
4. **Nudging Product Owner on Micromanagement** One to one discussion with PO on their goals & thought provoking questions make them self-realize that micro-management is counterproductive. Innovation & accountability can only be achieved if team feel empowered. Also educate them about the Golem and Pygmalion effects, and the impact they have on team performance.
5. **Sustainable Pace** Coach PO on the value of sustainable pace by working closely and making PO realize how employee satisfaction can increase productivity automatically & satisfaction can only come if employees are not overloaded.
6. **Support top Leadership to enable Safe Environment for teams** Safety is a basic human need and a key to unlocking high performance. People are afraid to make changes, afraid to voice their opinions and afraid of making mistakes. If you have a culture of fear, none of your fancy processes or practices will help you. Demonstrate impact of their practices & leadership style on teams by assessing **psychological safety** of teams.
7. **Enable face to face Conversations** There is an inevitably cultural difference and language barriers if teams are distributed & not communicating directly, it can create misunderstanding & confusion impacting trust. So, if teams are distributed, realize PO to invest in **Video conferencing equipment for face to face conversations** by making him understand agile principles & values.
8. **Make PO understand Estimates are not Deadlines** The development team forecast the work to be completed within a sprint, make PO realize that this estimate can change and he should take this into account.
9. **Bring in Information Radiators** Help the team use a visual Scrum Board to see their progress. It will help team to meet their commitments & sprint goals boosting PO trust in team.
10. **Limiting Work in Progress (WIP)** Support team in limiting their WIP by making them aware of techniques like Swarming, enabling team meet their commitments.
11. **Team Working Agreements** Build trust through team alignment by facilitating creation of “Working Agreements”.

12. **Addressing Cultural Gap** Cultural differences is a major roadblock if teams are distributed. To address cultural challenge, cultural working preferences tool can be utilized where team reflects their preferences. Thereafter compiling the scores onto one chart & looking at the differences with open discussion can help in bridging cultural gaps.

13. **Leverage the Power of Gamification** Using games enable PO to share his personal life with the team. I use to facilitate 2 Truths and a Lie game for this.

14. **Clarity on Project Goal/Vision** Informal Constellations approach can be used to bring clarity on project goals. Here asking simple statements to team members like *"I have clear project vision"* with an open conversation can do the magic. I have seen team members feel more connected and committed towards the project after this exercise.

To share the vision & goals, Objectives and Key Results (OKR) technique can also be used.

15. **Promote Culture of Servant Leadership** Help PO promote culture of Servant Leadership in place of Command & Control. Schneider model quadrants can help them identify their way of working. Thereafter SCARF model can be used to influence PO, making them realize the benefit of minimize threats and maximize rewards by explaining how the brain works under both the circumstances.

16. **More Collaboration** Spider diagram exercise can be used to map PO leadership style to team relationship (where team members draw the relationship between various stakeholders using thick, thin, dotted lines depicting their relationship) After that showing the results with open discussion, enables PO to build effective collaboration strategy which can build the foundation for trust.

**Trust & Transparency at all levels can enable teams to move from "Doing" to "Being Agile" I believe it is this 2T factor not talent, the foundation for building high performing teams.**

*What do you guys think??*

***

To read this article online with embedded hypertext links, go here:

**https://www.linkedin.com/pulse/2t-factor-being-agile-sourav-singla/**

# About Sourav Singla

**Sourav** Singla, an enterprise Agile Ninja Coach with TCS, leveraged agile practices by exploiting the fundamentals of Agility for fortune 500 companies enabling continuous delivery of the highest value. He is part of 1000 Strong TCS Agile Coach Community that is part of TCS Enterprise Agile vision.

He is a Certified Team Coach (CTC).

He has guided many large enterprises through agile transformations and helped them on their journey to continuous improvement in their product development capability by working closely with teams, programs & leadership, understanding their potential, challenges & strategic goals.

Sourav is passionate about helping teams to deliver immediate value and exceed expectations through building high-performing teams and implementing proven agile methods.

As a coach he seeks clarity around purpose, shared vision & values to bring alignment towards achievement, innovation, and sustainability.

He gives importance to people side of change & focus on bringing meaningful change into organizations that can improve the lives of people.

Possess deep experience in agile practices, leadership coaching, organizational evolution & scaling agile.

He is a strong believer of distributed agile model & has good experience in creating & leading agile teams in distributed setup.

He lives in Mumbai, India, you can reach Sourav on LinkedIn (https://www.linkedin.com/in/sourav-singla-safe-agilist-csp-icp-acc-csm-cspo-ssm-lssg-35b65924/)

# BDD and the four pillars of business agility

By John Ferguson Smart

Posted on June 14, 2018

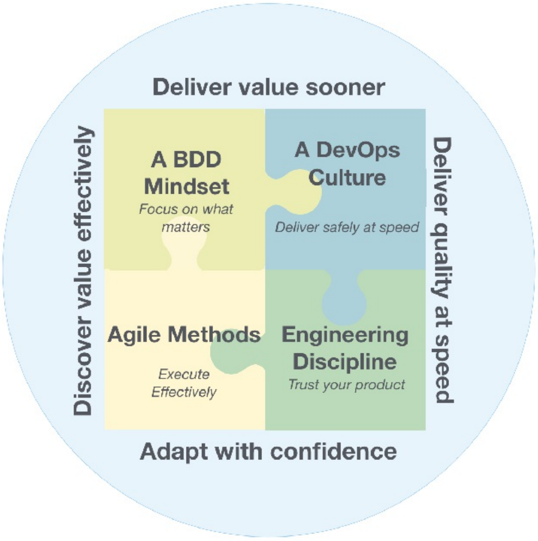
People often ask where Behaviour Driven Development, or BDD, fits with regards to other agile practices. Managers might say “BDD sounds great. But how do we align our agile transformation strategy with BDD adoption?”. Some ask “Does BDD work with Scrum/SAFe/Kanban/<insert your favourite agile methodology here>”? Some folk even simply ask “So are we doing BDD or Agile?”

To answer questions like these, this article tries to put BDD in the context of other agile practices.

**Behaviour Driven Development** (or BDD) is a collaborative approach that helps teams focus on delivering high value features sooner. BDD builds on and extends standard agile practices such as sprint planning and backlog grooming, user stories and acceptance criteria, and makes them much more effective.

Teams practicing BDD use structured conversations around business rules and concrete examples to build a deeper shared understanding of the problems they need to solve. They often automate the acceptance criteria they discover in the form of automated acceptance tests, or *executable specifications*, that provide both documentation about what the application does, and feedback about whether the application works.

But if you are trying to make a large organisation more agile, then BDD is just one part of the picture.



The Four Pillars of Business Agility

## The Four Pillars of Business Agility

In fact, there there are four key enablers, or pillars, to business agility:

- **Agile Methods**
- **A BDD Mindset**
- **Engineering Discipline**
- **A DevOps Culture**

**Agile Methods**, such as Scrum and Kanban, focus primarily on team organisation, coordination, and change management. Some would say that the value of a framework such as Scrum or Kanban is not so much to make you deliver faster, but to make it more obvious why you are not. To actually deliver faster, you need the other three pillars.

**A BDD Mindset** accelerates the flow of value and reduces waste and rework by enabling teams to discover and focus on the features that really matter ("building the right thing").

**Engineering Discipline** includes a range of practices and techniques that help us build more reliable, more maintainable code faster. Test Driven Development (TDD), Continuous Integration, Automated Acceptance Testing and Clean Coding all fall into this category. Such practices are essential if you want to deliver quickly and reliably.

Finally, **A DevOps Culture** gives teams the ability to deliver features safely to production at speed ("delivering the thing safely in time").

All of these areas overlap and interact. Requirements Discovery practices from BDD, such as Impact Mapping, Feature Mapping, and Example Mapping, play a key role in release and sprint planning and backlog grooming. And executable specifications, the flip side of BDD, rely heavily on solid engineering practices to keep the maintenances costs down and the feedback quick and reliable.

In sum, improvements in any of these areas in isolation are not enough to produce great results. You need a coordinated approach that address all four pillars if you want to get sustainable results.

## Acknowledgements

Model refined and refactored with the kind help of Peter Suggitt and Jan Molak. The first version of this model came out of a conversation with my good friend Marco Tedone some years ago, in a car in the mountains between Pune and Mumbai.

***

To read this article online with embedded hypertext links, go here:

**https://johnfergusonsmart.com/bdd-four-pillars-business-agility/**

# About John Ferguson Smart

**John** is an international keynote speaker, consultant, author and trainer well known in the Agile community for his many books, articles and presentations, particularly in areas such as BDD, TDD, test automation, software craft and team collaboration.

Author of the best-selling *BDD in Action* as well as many other books, John helps organisations and teams around the world deliver better software sooner through more effective collaboration and communication techniques, and through better technical practices.

Very active in the Open Source community, John also leads development on the innovative Serenity BDD test automation library, described as the “best open source selenium web driver framework”. John is also the founder of the Serenity Dojo, a set of specialised online training courses designed to help testers become experts in BDD and test automation.

# Want to do an Agile Transformation? Don’t. Focus on Flow, Quality, Happiness, Safety and Value

By Jonathan Smart

Posted on July 21, 2018

It was a Saturday morning. I was standing in line at a coffee shop, thinking about flow, as I often find myself doing when waiting in line. I reached for my phone to pass the time (while stuck behind a constraint in the cafe’s system of work) and sent the tweet above. This tweet went, relatively-speaking for me, viral, breaking in to triple digits of likes and retweets. It seems to have struck a chord.

> *If you want to do an Agile Transformation, don’t. Focus on Flow, Quality, Happiness, Safety & Value and you will end up transforming to have agility*

This is the first of a series of posts, where I’m going to share a number of **anti-patterns** and corresponding **patterns**. It is worth noting that everything is context dependent. An anti-pattern for one scenario might be a pattern in another context. That said, I believe that the anti-patterns to be presented are applicable in the majority of contexts.

This is based on lessons learnt through doing including learning from failing. Along with many talented people, we’ve been servant leaders on better ways of working (the application of agile, lean , DevOps, design thinking, systems thinking and so on) across a large (80,000 people), old (300+ years old), global, not-born-agile, highly regulated enterprise, with personal experience of delivering change with an agile mindset, principles and practices since the early 1990’s, about a decade prior to the Agile Manifesto, when the term ‘lightweight processes’ was used. The anti-patterns and patterns are also based on learnings from the community, from other horses (rather than unicorns) on similar journeys, as we are all at a turning point in the Age of Digital.

## Anti-pattern: Doing a capital ‘A’, capital ‘T’ Agile Transformation

A capital ‘A’, capital ‘T’ Agile Transformation, from the perspective of an employee, infers involuntary, mandatory change being done **to** you, whether you like it or not.

The capital ‘T’ denotes that you **have** to change and the capital ‘A’ denotes **how** you are going to change. Both of these words carry baggage.

Not surprisingly, this triggers fear and resistance for many reasons, including loss of control, uncertainty, changing habits, fear of failure, fear of incompetence, more work, change fatigue and ‘better the devil you know’.

by Ross Findon on Unsplash

From an evolutionary perspective, depending on the messaging of the **why** and depending on **how** the change is approached, and in particular for those with a fixed mindset, change drives a fear of survival, which leads to resistance and less rational thought as the primitive brain takes over.

Can I change? What if I can’t adapt? Will I still be able to pay the bills?

Looking at autonomy, purpose and mastery, as per Daniel Pink's Drive and that human motivation is primarily intrinsic for knowledge work, two of the top three motivators have been taken away. There is a lack of autonomy (you have to do this thing) and a lack of mastery (you're a beginner again, possibly after a long career). If the **why** is articulated as cost reduction or profitability, meaningful purpose is also removed, taking away all three categories of human motivation.

> *"The problem with the amygdala and its fight-or-flight response today is that it sets off alarm bells whenever we want to make a departure from our usual, safe routines. The brain is designed so that any new challenge or opportunity or desire triggers some degree of fear. Whether the challenge is a new job or just meeting a new person, the amygdala alerts parts of the body to prepare for action — and our access to the cortex, the thinking part of the brain, is restricted, and sometimes shut down."* (One Small Step Can Change Your Life. The Kaizen Way, Dr Robert Maurer, 2014)

This evolutionary fear of change is also seen in loss aversion, which is people's tendency to prefer avoiding losses to acquiring equivalent gains. This evolutionary tendency to loss aversion further cements people's desire to maintain the status quo.

> *"Humans may be hardwired to be loss averse due to asymmetric evolutionary pressure on losses and gains: for an organism operating close to the edge of survival, the loss of a day's food could cause death, whereas the gain of an extra day's food would not necessarily cause an extra day of life."* (Loss aversion, Wikipedia)

Agile should not have a capital A unless it is at the start of a new sentence. Agile is not a noun. It is not a trademark. You can't buy Agile In A Box (really, you can't). By the very fact that agility is optimal in complex contexts, where the What and the How is inherently unknowable in advance, where acting in the space changes the space, there cannot be a one size fits all solution. It is about exhibiting agility, being agile not doing Agile. It is first and foremost a mindset that informs every single deliberate decision and automatic reaction.

> ***"If the path ahead is clear, you are on someone else's path", Joseph Campbell***

The capital 'A' Agile mindset leads to the Agile Industrial Complex. It leads to a number of anti-patterns, such as the imposition of practices on people and to certification schemes with questionable value. We don't want agile for agile's sake or DevOps for DevOps sake. This can lead to local optimisations where the expected business benefits, end to end, do not materialise.

There is a need to focus on Why are we doing this and what the desired business outcomes are. Then agile, lean, DevOps, Design Thinking and so on are bodies of knowledge, they are tools in the toolbox, to achieve those outcomes, applying what works in your unique context and continually improving through experimentation.

### Pattern: Start with Why and focus on Outcomes

First, as well articulated by Simon Sinek, start with why. There should be a clear and well communicated **Why** for the organisation of the need to change ways of working, why constant improvement is needed, with nuanced context-sensitive and relevant definitions of why for the any sub-organisations within the parent organisation which may have their own cultural norms, history, folklore and legacy ways of working.

by Gaelle Marcel on Unsplash

The 'why' should be more than profitability, shareholder returns or stock price. As per the article 'The Irrational

Side of Change Management'

*"what motivates leaders does not motivate most of the employees"*

Research shows that when employees are asked what motivates them the most in their work they are equally split across five forms of impact: (1) society (2) customer (3) company (4) team (5) the individual.

> *"What the leader cares about (and typically bases at least 80 percent of his or her message to others on) does not tap into roughly 80 percent of the workforce's primary motivators for putting extra energy into the change program."*

Teal organisations, the most evolved, are driven by a higher level purpose. This is echoed in 'Drive' by Dan Pink, where people are motivated by a transcendent purpose. Ensure that the definition of why has a higher level purpose and covers society, customers, company, team and the individual.

From there, identify high level, thematic desired **outcomes**. Start with what does awesome look like, what is the current reality and what are the obstacles. Next, derive, prioritise and theme outcomes which get you to your state of awesomeness.

For us, our desired outcomes are described as **Better Value Sooner Safer Happier**, each of which is measurable.

- Better => Quality => Production Incidents, Resilience, Static code analysis measures
- Value => context specific unique measures per quarterly Business Outcome (e.g. customer Net Promoter Score increased 10 points, reduced carbon usage by 10%, increased gender diversity by 15%, increased lending to small and medium firms by £100m, reduced Risk Weighted Assets held on balance sheet by 20%, etc.)
- Sooner => Flow => Lead Time, Throughput, Release Cadence, Flow Efficiency. Be wary of becoming a feature factory. Reduce Lead Time and spend more time with the end users, refactoring and innovating.
- Safer => GRC Control Compliance (e.g. InfoSec, Know-Your-Client, Data Privacy, GDPR type mandatory requirements). Speed & Control. Agile not fragile.
- Happier => increased customer and colleague satisfaction, via surveys and feedback loops

Ultimately, it is all about Flow. This needs balancing measures, as anything can be done badly, so that it is not Flow at the expense of quality, happiness, safety or value.

On a regular cadence, a one pager is sent to senior leaders with measures for the above, including vector measures, the rate of improvement. Everyone can see everyone else's data. In addition, a real time dashboard with drill down is available.

The scope is the whole organisation, to enable organisational agility. It is Incremental and Disruptive, Exploit and Explore. It is equally aiming to delight customers in existing markets and to delight new customers in new markets.

Then, for the reasons above, rather than doing a capital 'A', capital 'T', Agile Transformation for the sake of agile and framing it as such, and commence the rollout of a set of prescriptive practices or a prescriptive framework, frame the change around the **Why** and around the improvement in the identified **Outcomes**.

This appeals to intrinsic motivation, it is asking people to bring their brain to work, to understand and internalise the mission, and in an empowered manner to take personal ownership for working out **how** to achieve the desired outcomes. Agile principles and practices, Lean, DevOps, Systems Thinking, Design Thinking and so on are tools in the toolbox, and are supported with training, coaching and psychological safety in a context sensitive, not one-size-fits-all, pull-based not push-based manner.

***

To read this article online with embedded hypertext links, go here:

**https://medium.com/sooner-safer-happier/want-to-do-an-agile-transformation-dont-focus-on-flow-quality-happiness-safety-and-value-11e01ee8f8f3**

# Want to scale agile? Don’t. Descale the work first. Achieve big through small.

By Jonathan Smart

Posted on August 18, 2018

Photo by chuttersnap on Unsplash

This is the second post in a series, sharing a number of observed **anti-patterns** and corresponding **patterns** on the topic of organisational agility (aka digital transformation).

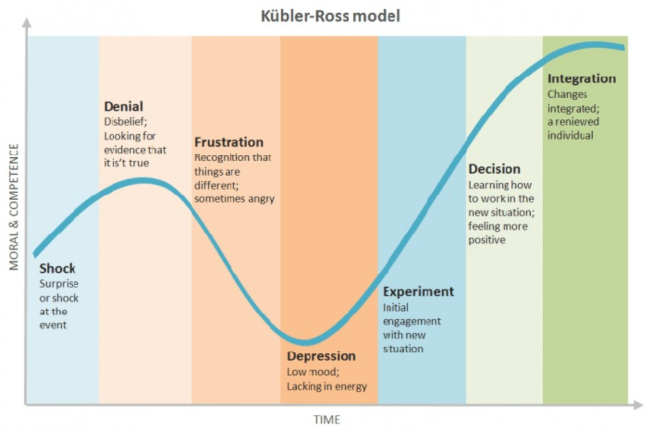
We are in the midst of a Turning Point in a 50 year cycle, in the Age of Digital, as well articulated by Carlota Perez in ‘*Technological Revolutions and Financial Capital*’. At the time of writing, seven of the top ten firms by market capitalisation are technology companies. Less than two months ago, on 26th June 2018, General Electric, the last remaining original constituent of the Dow Jones index which was created in 1896, left the index, as it was contributing less than half a percent. In 2004, GE was the largest firm in the world by market value and only two years ago, in 2016, GE was still in the top ten. This is an indication of how we are in the Turning Point, how the previous industrial incumbents are shrinking and how firms with new business models and new ways of working, leveraging significant shifts in technology have become the new dominant forms of human organisation.

In sharing the anti-patterns and patterns, it is worth noting that everything is context dependent. An anti-pattern for one scenario might be a pattern in another context, in particular based on the current cultural norms of an organisation. That said, I believe that the anti-patterns to be presented are applicable in the context of the majority of large, old, bureaucratic, global enterprises (horses rather than unicorns).

They are based on lessons learnt through doing including learning from failing. Along with many talented people, we’ve been servant leaders on better ways of working (the application of agile, lean , DevOps, design thinking, systems thinking and so on) across a large (80,000 people), old (300+ years old), global, not-born-agile, highly regulated enterprise, with personal experience of delivering change with an agile mindset, principles and practices since the early 1990’s, about a decade prior to the Agile Manifesto, when the term ‘lightweight processes’ was used. The anti-patterns and patterns are also based on learnings from the community, from other horses (rather than unicorns) on similar journeys.

## Anti-pattern: The bigger the capital ‘T’ Transformation, the bigger the change curve

The **Kubler-Ross Curve** originated from psychiatrist Elisabeth Kubler-Ross’s work on grief, published in 1969. It has been found to be valid in the majority of situations relating to change and we have repeatedly observed this pattern to hold true via feedback from colleague surveys.


Kübler-Ross model
MORAL & COMPETENCE
TIME
Shock
Surprise or shock at the event
Denial
Disbelief; Looking for evidence that it is't true
Frustration
Recognition that things are different; sometimes angry
Depression
Low mood; Lacking in energy
Experiment
Initial engagement with new stuation
Decision
Learning how to work in the new situation; feeling more positive
Integration
Changes integrated; a reniewed individual

The bigger the capital ‘T’ Transformation, the bigger the change curve. If embarking on one large, broad, Transformation, expect an almighty big and deep dip in the curve. The bigger the dip, the harder it is to climb out of and the longer it takes. Given that some firms are facing an existential threat, there may not be sufficient time to climb out of that big dip.

In large, diverse, regulated, multinational organisations, where the cultural norm is most likely to be control or competence based, a capital ‘T’ Transformation with a big dip in the curve, will make the journey a harder and more challenging one. There will be a greater degree of denial, frustration and anger. The change stands a higher chance of cultural tissue rejection, with more ammunition for those averse to change. Things will get significantly worse before they get better.

For organisations that take this approach, with a broad scope, where people leave as a Project Manager on a Friday and rejoin as a Scrum Master on Monday, and in some cases need to reapply for their new role, the chances of genuine, embedded, internalised, long lasting, successful change which leads to improved business outcomes, over cargo cult behaviours and new labels on existing ways of working, are greatly lowered. As per this post , this drives fear, which drives a lack of action and resistance.

## Hardest at the beginning

When starting out on transforming ways of working, increasing organisational agility, embarking on a ‘digital transformation’, it is hardest at the beginning. The antibodies to change are strong, there are vested interests at stake, the impediments to better ways of working and flow will be at the their highest, there will be the most amount of dependencies impeding flow, understanding is low, the force is strong with cognitive biases with no anecdotal stories or hard data from the organisation to challenge them and this is Yet Another Transformation (time to put your head in the sand and let it blow over). An analogy is skiing. It’s cold, painful, slow and hurts when learning to ski. It’s hardest at the beginning, uses up the most energy, when snow ploughing. Break past that, and once able to parallel turn, enjoyment goes up, speed goes up, energy usage for the same distance and time goes down, it becomes fun and addictive. Does it make sense to have a large number of people all snow-ploughing at the same time, on poor quality snow, in an environment not yet set up for skiing, bumping into each other, without enough ski instructors to go around?

## Limited re-learning velocity

Furthermore, large, old, bureaucratic, traditional organisations have a limited capacity and a tolerance with which they can change over time. Organisations have a limited re-learning velocity. Re-learning requires organisational unlearning, which is harder than learning from a clean sheet. Behavioural science studies show that cognitive overload is a major theme in rejection of change. Cognitive reasoning is finite and easily depleted. According to research by

Dr Wendy Wood, approximately 40% of decisions that people make everyday are not decisions, but are habits and the majority of these habits in going from a traditional to a new way of working, need to be unlearnt.

> *"The thoughtful intentional mind is easily derailed and people tend to fall back on habitual behaviors. Forty percent of the time we're not thinking about what we're doing. Habits allow us to focus on other things… Willpower is a limited resource, and when it runs out you fall back on habits."* (source)

Where the change has not been internalised and embedded, where it is forced across an organisation in a broad manner, it is like one large elastic band, as soon as a leader mandating capital 'T' Transformation moves on, the organisation (people's habits and codified processes) snap back into previous ways of working. It takes 3 to 5 years, best case, for a large traditional organisation to develop a new muscle memory. And then there is no end date to continuous improvement. For more on this topic, see Barry O'Reilly's book Unlearn (Nov 2018).

## Big Bang. Big Risk.

This approach is also not living it's own values or applying its own principles. It is not applying an agile mindset to increasing organisational agility. It is big batch, big bang and big risk. It is approaching change in a manner counter to the change being asked of colleagues.

## Bureaucracy

According to Parkinson's Law, *"work expands so as to fill the time available for its completion"*.

I propose a corollary to this which is that *"processes, control points and standards expand based on the number of audit and control staff employed"*. This is in an uncontested space (i.e. in the absence of a group of people focussed on optimal ways of working)

Parkinson goes on to say that *"the number employed in a bureaucracy rise by 5–7% per year irrespective of any variation in the amount of work (if any) to be done"*.

He cites two factors: (1) *"An official wants to multiple subordinates, not rivals"* and (2) *"Officials make work for each other"* and he gives the British Colonial Office as an example. In the nearly twenty years from 1935 to 1954, looking only at peacetime years, the average rate of growth of employees at the Colonial Office was 5.89% each year (within a narrow range of 5.24% to 6.55% p.a.), whilst the British Empire shrunk by 76% from 17 to 4 million square miles in the same time period. The size of the Colonial Office was inversely proportional to the size of the the Empire.

> *"It would be rational, prior to the discovery of Parkinson's Law, to suppose that these changes in the scope of Empire would be reflected in the size of its central administration. But a glance at the figures shows that the staff totals represent automatic stages in an inevitable increase.* ***And this increase, has nothing to do with the size — or even the existence — of the Empire.****"* (Cyril Parkinson, 1955)

As large, old, enterprises, with years of organisational scar tissue and 100 year old ways of working are going to be at the highest level of inefficiency and bureaucracy that their revenues can (or perhaps in the Age of Digital, cannot) support, this is one reason why it is sub-optimal to take that organisation and apply a scaled agile framework across the whole organisation in one go.

It is important to descale the organisation, to descale the work, before scaling agility (not scaling capital 'A' Agile, which is doing not being).

> *Achieve big through small*

Start with small teams, small slices of value and small investments.

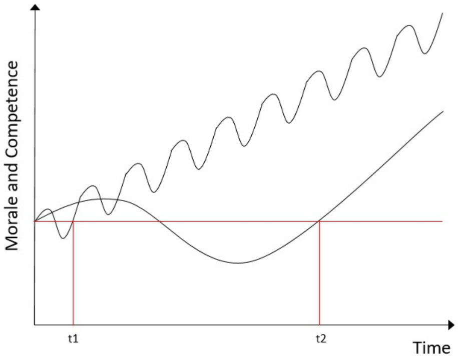
> *"Nail it before you scale it"* (thanks Cliff Hazell)

# Pattern: In order to scale, descale first. Achieve big through small

Photo by Serhat Beyazkaya on Unsplash

Instead of a big bang transformation, with one big dip in the curve, achieve a big outcome through early, often and small slices of value.

Pursue evolutionary and continuous transformation aligned to outcomes, linking together a series of smaller change curves. Start in areas which are naturally receptive, the natural champions. The dips are not as deep, the learning and feedback is quicker, there is less risk and the champions, who have been trying to do this despite the firm in the past, are best placed to beat a path through the organisational jungle, likely with a growth mindset and personal resilience.



This approach is in line with the Kanban Method principle “Agree to pursue incremental, evolutionary change”. As time goes by, I have found myself to value David J Anderson’s Kanban Method principles and practices more and more, in the context of organisational agility, at all levels from strategy to the team.

Hence, don’t take a large team working in a traditional manner, on a traditionally developed product (whether IT or not) and apply revolution all in one go. Apply an agile mindset to the rollout of agile. Achieve big outcomes through (1) small teams, (2) small investments and (3) small slices of value, supported with capability building, training and coaching as well as help to remove organisational impediments. Identify where there is ‘elephant carpaccio’ which can be delivered, when starting out with a traditional waterfall team, likely as per Conway’s Law, with a monolithic system. Eventually there should be no elephant in the room. Leaders should ensure that there is a psychologically safe environment for experimentation and learning.

# Case study of descaling work

By way of an example, I’m aware of a scenario where there was a team of about 100 people who had multiple multi-year failed attempts to deliver business value in a waterfall manner. Following this, with the appointment of a leader who had a track record of successful delivery, a team of five was created, working with agile principles and practices. Within 12 weeks, there was a product in the hands of customers, in a production environment, solving a customer need and providing much needed learning and feedback. From this point on, the team size didn’t grow to be above three teams of nine or fewer people each, despite expanding significantly in scope in terms of the business lines supported, due to the success in the first business line.

Taking the original 100 people and applying a frog march of mandated certification, re-applying for your job with a different title and a cookie-cutter approach would not have addressed the inherent bloat at the time (it's not a 100 person problem, in fact 100 people is a large part of the problem), would not have internalised the change so that it comes from within thus building a learning organisation, is not optimised to context, is costly and does not maximise business outcomes & customer delight.

## Fractal

For large organisations, and the approach that we have taken, where there are multiple business units (each one large enough to be standalone company in it's own right, and in fact used to be, with their own culture and folklore), the 'achieve big through small' approach can be taken concurrently (whilst limiting Work In Progress) in a fractal manner. Each business unit pursues a limited series of small change experiments starting in fertile soil, with a small central Centre of Enablement (CoE) providing servant-leadership support and dealing with bubbled up organisational impediments. An agile mindset is applied in terms of empowerment and not being prescriptive on the How (within guardrails and a common vocabulary). This in turn can be done at the sub-BU level, and so on. The leads from each business unit come together weekly as a virtual team. Applying an agile mindset, all of the virtual teams of BU or sub-BU leads are agile team sized (single digits). It's quick to spot common organisational constraints and allows for swarming on alleviating the constraints.

## Scale agility, not Agile, vertically then sideways

A mistake that we've made in the past has been to start at the team level and go sideways (more teams), along with the top level support. However, even with specific targeted training, this can fail to engage one or more levels of middle management, also known as the Frozen Middle or more kindly, the Pressurised Middle, as there is not always a clear role to play in the change. This is a common characteristic in any culture change in large organisations and is not a reflection of the people, it is a reflection of the situation they are in and how people in these roles are engaged.

A personal learning, when starting small, from a people perspective, is to have a vertical slice of an organisation go first. The leadership team is team #1. With the existing structure, have a vertical slice of that org volunteer to go first, including leaders at all levels, preferably natural champions, with as few dependencies on other teams as possible. Ideally this will be value stream / product / service aligned, not a role-specialisation alignment, such as just the BAs or just the Engineers or just the PMs. Middle management at as many levels as there are, have an explicit role, coaching and being coached on continuous improvement in ways of workings, as per the Toyota Coaching Kata and with multiple-level portfolio Kanban being adopted to focus on visualising and limiting work in progress.

Then scale agility (not capital 'A' Agile) sideways, at a sustainable pace, working towards the organisation becoming a network of interdependent services. Go for more slices of the organisation, which are value stream / product / service aligned, supported by a small BU Centre of Enablement, providing coaching, training, shared learning, clearing the path for the teams. Pursue incremental, evolutionary, outcome-oriented, continuous transformation.

This doesn't mean that scaled agile frameworks are not used. It is up to each team and area to decide what works best for them. Each framework is a valuable body of knowledge and can be a good departure point. In some cases they can provide a common vocabulary and in all cases it's about trying what works in your unique context, with a focus on better business outcomes (as per my previous post). We aim to avoid framework fundamentalism, preferring to be Omnists. For more on this topic, see Dan North's excellent article on SWARMing.

Again, at the beginning it's the hardest, as the rest of the organisation is not set up for Continuous Everything. Several years in and good progress has been made by supporting functions (GRC type functions, InfoSec, Compliance, Audit and so on) also working in a way which supports small and often, conversation over a contract, with multidisciplinary long lived small teams, value stream aligned and in a context-relevant not one-size-fits-all manner.

## Exemplar Community

To help amplify the learning, the overcoming of impediments, and adoption of better ways of working, we have an Exemplar Community. There are benefits of membership, such as additional training, external speakers, shared learning and prioritisation on the pull of the virtual andon cord. You don't need to be exemplary to join, it is voluntary and there is a psychological contract in that teams agree to strive to become exemplary, jumping in with both feet, with a focus on measurable business outcomes (not activity or output). Consistently these teams exhibit far superior outcomes (for example 23x fewer production incidents on average), provide hard data as to benefits for the critics who are swayed by hard data, provide storytelling for emotional buy in and via cellular mitosis are able to spread better ways of working.

## UK Government Digital Service (GDS)

UK GDS have taken a similar path. To quote Adam Maddison, ex-Head of Agile Delivery at GDS from Agile Cambridge 2016:

*"What's our approach to scaling at GDS? Do not scale. But we do have a number of large programmes."*

*"At GDS we absolutely haven't ignored scaling frameworks. We're quite happy to use features from those frameworks if they truly deliver value. But we are absolutely not wedded to any methodology or framework. We will always adapt features to our own use."*

In both the case of GDS and in our context, the 'big' in 'achieve big through small' is articulated via the Roadmap, which articulates Strategic Objectives and quarterly Business Outcomes on long lived value streams / products / services. These in turn are broken down into small vertical slices of value, like layers of an onion, as outcomes not activities and delivered via Continuous Everything. Achieving big outcomes through many small steps and continuous learning.

## Summary

- Achieve big through small
- Descale the organisation and the work first
- Nail it before you scale it
- Scale agility, not capital 'A' Agile, at a sustainable place, with support
- Leaders go first
- Work towards the organisation becoming a network of interdependent long-lived services, with high cohesion, low coupling and customer-centricity
- Apply an agile mindset to the adoption of agility

***

To read this article online with embedded hypertext links, go here:

**https://medium.com/sooner-safer-happier/want-to-scale-agile-dont-descale-the-work-first-achieve-big-through-small-7bdd09962540**

# Organisational Agility: Give People a VOICE

By Jonathan Smart

Posted on September 28, 2018

**Don’t Mandate One Way™.**

Photo by Aaron Burden on Unsplash

> “You are special, you are a beautiful and unique snowflake”

You are, your team is and so are the complex adaptive systems that you are part of.

This is the third article in a series, sharing a number of observed **antipatterns** and corresponding **patterns** on the topic of organisational agility.

We are entering a new Deployment Period in a 50 year cycle, in the Age of Digital, as well articulated by Carlota Perez in ‘Technological Revolutions and Financial Capital’.

Traditional organisations need to adopt new ways of working in order to survive and thrive, to leverage the on-demand compute, information and communication power which is in the cloud and in our pockets, to keep up with competitors and challengers, and to delight customers. Today, for complex knowledge work, many are still applying a management model from the early 1900s optimised for manual labour in factories.

The antipatterns and patterns are based on lessons learnt through doing, including learning from failing. We have been servant leaders on better ways of working, applying agile, lean, DevOps, design thinking, systems thinking and so on, across a large (80,000 people), old (300+ years old), global, not-born-agile, highly regulated enterprise. I have been delivering change with an agile mindset, principles and practices since the early 1990s, about a decade prior to the Agile Manifesto, when the term ‘lightweight processes’ was used. The antipatterns and patterns are also based on learnings from the community, from other horses, rather than unicorns, on similar journeys.

## Antipattern: Golden Hammer

The mandating of one set of prescriptive practices, organisation-wide, is often combined with the capital A Agile and capital T Transformation antipatterns. The Golden Hammer antipattern was summed up by Abraham Maslow in 1966 who said, *“I suppose it is tempting, if the only tool you have is a hammer, to treat everything as if it were a nail”*. Herein lie two antipatterns: (1) Golden and (2) Hammer.

# Antipattern part 1: Golden

Mandating any one set of practices on people and teams is an antipattern.

> *"Imposing a process on a team is completely opposed to the principles of agile, and has been since its inception. A team should choose its own process — one that suits the people and context in which they work. Imposing an agile process from the outside strips the team of the self-determination which is at the heart of agile thinking. Not only should a team choose their own process, the team should be in control of how that process evolves."*, Martin Fowler, 2006 (source)

The Agile Manifesto emphasises individuals and interactions over processes and tools, collaboration over contract negotiation, motivated individuals who are given support and trusted to get the job done, self organising teams regularly reflecting on how to be more effective, tuning and adjusting their behaviour, practices and processes.

> *"With so many differences, how can we say there's one way that's going to work for everybody? We can't. And yet what I'm hearing so much is the Agile Industrial Complex imposing methods upon people, and that to me is an absolute travesty. I was gonna say "tragedy", but I think "travesty" is the better word because there is no one-size-fits-all. "*, Martin Fowler, Agile Australia 2018 (source)

Unfortunately, it is common at the moment to see prescriptive processes being mandated across teams and across organisations. This should not be done. Inflicting capital 'A' Agile is not empowering, it does not show respect for people, it drives fear and resistance and it is not taking an agile approach to agility. It moves the locus of control to be external, reducing psychological ownership and intrinsic motivation.

Perhaps this is not surprising where many leaders in large, traditional firms have got to where they've got to with a reductionist, predictive, command & control mindset. This gives rise to the Agile Industrial Complex. An old ways of working view is being applied to new ways of working. In some cases, it may take a new generation of leaders.

> "*Knowledge workers themselves are best placed to make decisions about how to perform their work*", Peter F. Drucker

## Antipattern part 2: Hammer

There is no one-size-fits-all for organisational agility. There is no One Way™ that optimises outcomes in all contexts.

Your organisation, your customers, your value propositions, your environment, your processes, your system of work, your leaders, your teams, your constraints, your starting point, your behavioural norms, your heritage, your brand, your team and you, are unique.

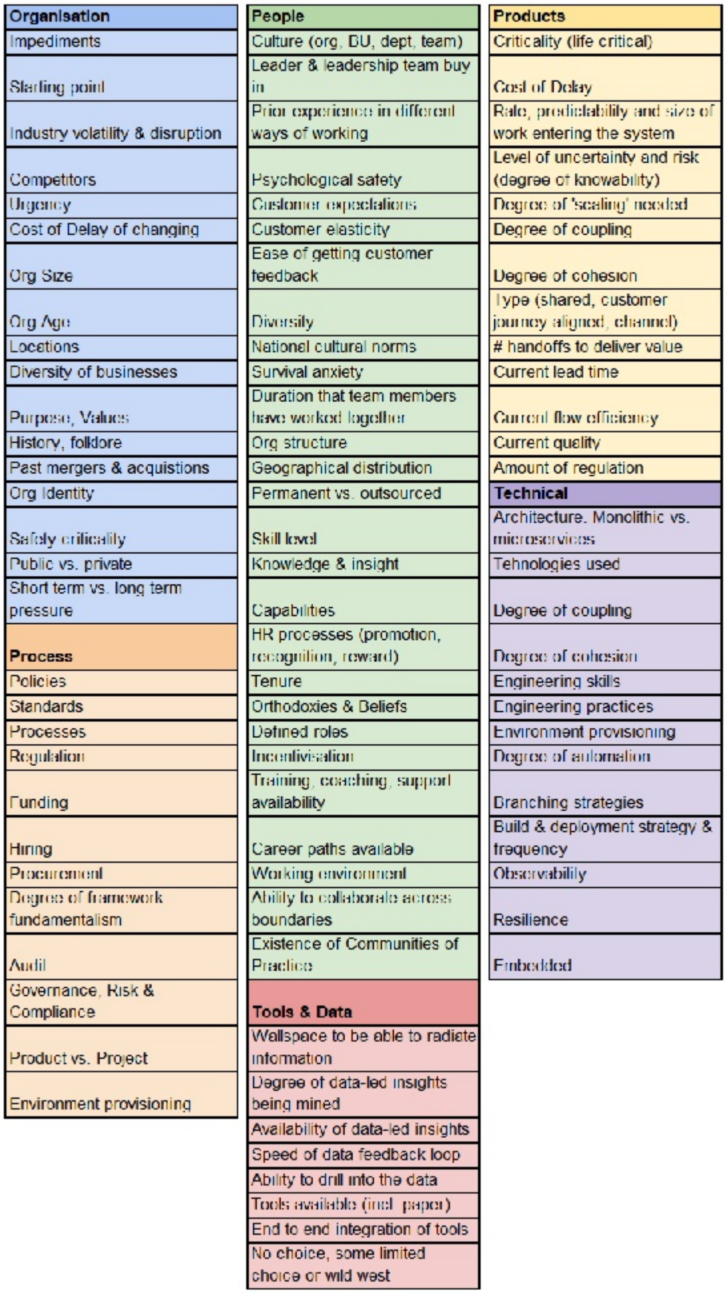
> *"If the path ahead is clear, you're on someone else's path"*, Joseph Campbell

Organisations are heterogeneous, not homogeneous. Organisations are emergent, not predictive. Organisations are complex adaptive systems.

> *"A complex adaptive system is a dynamic networks of interactions, their relationships are not aggregations of the individual static entities, i.e., the behavior of the ensemble is not predicted by the behavior of the components. They are adaptive in that the individual and collective behavior mutate and self-organize corresponding to the change-initiating event."* Wikipedia (source)

Growing agility (rather than scaling capital 'A' Agile) enterprise-wide is about leveraging that complexity, diversity and emergence, not inflicting sameness on every team. The approach to organisational agility cannot be cookie-cutter.

Your context is unique. To illustrate this point, here are some of the factors that make up your context:



**Organisation**

- Impediments
- Starting point
- Industry volatility & disruption
- Competitors
- Urgency
- Cost of Delay of changing
- Org Size
- Org Age
- Locations
- Diversity of businesses
- Purpose, Values
- History, folklore
- Past mergers & acquistions
- Org Identity
- Safety criticality
- Public vs. private
- Short term vs. long term pressure

**Process**

- Policies
- Standards
- Processes
- Regulation
- Funding
- Hiring
- Procurement
- Degree of framework fundamentalism
- Audit
- Governance, Risk & Compliance
- Product vs. Project
- Environment provisioning

**People**

- Culture (org, BU, dept, team)
- Leader & leadership team buy in
- Prior experience in different ways of working
- Psychological safety
- Customer expectations
- Customer elasticity
- Ease of getting customer feedback
- Diversity
- National cultural norms
- Survival anxiety
- Duration that team members have worked together
- Org structure
- Geographical distribution
- Permanent vs. outsourced
- Skill level
- Knowledge & insight
- Capabilities
- HR processes (promotion, recognition, reward)
- Tenure
- Orthodoxies & Beliefs
- Defined roles
- Incentivisation
- Training, coaching, support availability
- Career paths available
- Working environment
- Ability to collaborate across boundaries
- Existence of Communities of Practice

**Tools & Data**

- Wallspace to be able to radiate information
- Degree of data-led insights being mined
- Availability of data-led insights
- Speed of data feedback loop
- Ability to drill into the data
- Tools available (incl. paper)
- End to end integration of tools
- No choice, some limited choice or wild west

**Products**

- Criticality (life critical)
- Cost of Delay
- Rate, predictability and size of work entering the system
- Level of uncertainty and risk (degree of knowability)
- Degree of 'scaling' needed
- Degree of coupling
- Degree of cohesion
- Type (shared, customer journey aligned, channel)
- # handoffs to deliver value
- Current lead time
- Current flow efficiency
- Current quality
- Amount of regulation

**Technical**

- Architecture. Monolithic vs. microservices
- Tehnologies used
- Degree of coupling
- Degree of cohesion
- Engineering skills
- Engineering practices
- Environment provisioning
- Degree of automation
- Branching strategies
- Build & deployment strategy & frequency
- Observability
- Resilience
- Embedded

Contextual Criteria

It is interesting to note that the People category has the most contextual criteria and Tooling the least, reinforcing how much this is about people.

It was a quick exercise to brainstorm these contextual criteria and it surprised me that as many as 90 criteria emerged in a short amount of time. This is 1.2 x $10^{27}$ (1.2 octillion) unique combinations of the criteria, assuming the criteria are binary, which they aren't, there are many more combinations. Clearly, your context is unique and across a large organisation there are many unique contexts, like fingerprints. And your context is changing. And that pace of change is getting faster.

There are two contextual criteria in particular that I'd like to call out. Scaling Agile and Culture.

**Scaling Agile** can have many interpretations and meanings, from a handful of teams on one large product with dependencies not broken, all the way to the diversity and complexity of organisation-wide agility with thousands of value streams, teams and contexts. Depending on why you want to 'Scale Agile' and what you mean by 'Scaling Agile', your approach will differ.

**Understanding the Incumbent Culture** is key. Within one large organisation there will likely be a number of separate prevalent cultures, due to history, folklore, location, leadership style and so on. It would be foolish to take one approach across different cultures. There are many cultural models to use as a guide. Take Westrum's typology as one example. If the starting culture is Pathological, then psychological safety will be thin on the ground. An autocratic culture is prevalent and in the worst cases there is a culture of fear with learned helplessness. I've spoken to teams who didn't dare to inspect and adapt. A revolutionary approach to changing ways of working is not likely to be successful in this context. I have seen it fail with the imposition of Scrum and synchronised Sprints, which was a case of inflicting revolution in an environment with a lack of safety. A better approach in this context is inviting evolution over imposing revolution, starting with what you do now, for reasons I go into below.

> *"The level of consciousness of an organisation cannot exceed the level of consciousness of its leader"*, Frederic Laloux

If the starting culture is Generative, if there is psychological safety, if there are already pockets of mastery, if there is strong survival anxiety with a high cost of delay, then it is possible that inviting revolutionary change is preferable in this context.

# Frameworks

The challenge for any framework of practices in an emergent context is the balancing line between prescription for beginners and flexibility as people gain mastery. A framework either explicitly caters to adaptability to context by not being prescriptive on practices, shining a light on the problems and letting people work out how to solve them, or it is prescriptive and dogmatic. In the latter case as mastery grows you can treat the framework as a departure point and inspect & adapt away from it, so it is no longer being 'done', if that improves your outcomes.

Scrum and its scaled variants such as SAFe, LeSS, Nexus and Scrum@Scale, are revolutionary rather than evolutionary as they introduce new roles, artifacts, events and rules. The new roles, artifacts and events are non-negotiable. You're either doing it or you're not doing it. The scaled variants have evolved in in a context of scaling vertically, many teams on one large product, with dependencies. Disciplined Agile supports context sensitive choice with guidance. It suits the context of enterprise agility and scaling horizontally, catering to diversity and emergence, with guardrails. The Kanban Method is evolutionary, you apply it to your process, it shines a light on flow and impediments to flow. More on this topic in a subsequent post.

All are useful tools to have in the toolbox. One size does not fit all.

---

# Pattern: Give People a VOICE

Don't mandate one set of prescriptive practices across your organisation. Instead, apply an agile mindset to organisational agility and **give people a VOICE**.

> ***V****alues & Principles*
>
> ***O****utcomes*
>
> ***I****ntent Based Leadership*
>
> ***C****oaching*
>
> ***E****xperimentation*

Photo by Ian Schneider on Unsplash

Have your own clear and updated **Values & Principles** that guide every decision across contexts. Focus on **Outcomes** (for example, Better Products Faster Safer Happier), with a clear Why, with measures and fast feedback loops, not agile for agile's sake or because your competitors are doing it. Practice **Intent Based Leadership** and empowerment based on the principle of Commander's Intent, decentralise decision making, strive for high autonomy with high alignment, including teams working out for themselves how they will improve on the Outcomes in their context. Agility by invitation, not Agile Imposition. Leaders go first, role modelling the desired behaviours, with a clear employee engagement model. Provide **Coaching** and support to unlearn old habits and learn new ones, to grow mastery and to remove organisational impediments. Leverage many bodies of knowledge, have many tools in the toolbox. Be an omnist. Leaders at all levels coach and support continuous improvement and technical excellence. Perform **Experimentation** at all levels with fast feedback. Probe, Sense and Respond at the organisational, business unit and team level. Implement double loop learning. Become a learning organisation. Be the best at being better.

## Values and Principles

Determine, update and repeatedly communicate the **Value and Principles** for your organisation, which inform every decision. These are the behavioural guardrails. They signal intent and they allow for behaviour which is not in line to be called out. This includes teams holding their leaders behaviour to account, helping teams to manage up. They should be timeless (for example, 'Customer Obsession' or 'Continuous Improvement') and apply across contexts.

Having defined your Principles, with many unique contexts, the practices will differ. To quote Dan North:

> ***Practices = Principles(Context)***

Practices will emerge by applying your unique Context to your Principles, with coaching and experimentation, leveraging many bodies of knowledge.

## Outcomes

As articulated in the first post in this series, for the organisation clearly articulate **Why** you are looking to improve ways of working and what the desired **Outcomes** are (for example, Better Products Faster Safer Happier), each one having one or more measures. The Agendashift From Obstacles To Outcomes (FOTO) exercise is one tool which can help identify the Outcomes. Another source of inspiration are the Capabilities listed in Accelerate (Forsgren, Humble, Kim, 2018) which are correlated to higher organisational performance.

## Intent Based Leadership

Adopt **Intent Based Leadership** as modelled by modern military command. Having provided the high alignment, give people high autonomy, empowering teams to improve on the Outcomes how they see fit, with fast feedback, data and support, starting small. Move authority to the information, not information to authority, decentralising decision making. Do not impose prescription. Make improvement of the desired Outcomes transparent. Adopt a pull, not push approach, within guardrails so that it is agile not fragile. Critically, foster psychological safety, so that people feel safe to experiment, moving the locus of control to be internal. Be a Transformational Leader, moving away from a command and control leadership style from the 1900s in the context of manual labour in a predictive environment. As per the State of DevOps Report, Transformational Leadership is highly correlated with higher organisational performance.

To engage colleagues in change, there are engagement models such as Agendashift or OpenSpace Agility which can assist, such that it is participatory, inclusive and appeals to intrinsic motivation.

> *"If you want to build a ship, don't drum up people to collect wood and don't assign them tasks and work, but rather teach them to long for the endless immensity of the sea."*, Antoine de Saint Exupéry

# Coaching

Provide **Coaching and support** on agility, not just on one prescriptive framework. Provide coaching and support on a high bar of technical excellence, on removing organisational impediments to flow and on continuous improvement both small (Kaizen) and radical (Kaikaku).

A federated, fractal structure of small Enablement Teams is a pattern which we have had success with. They perform a Servant Leadership role. Servant: there to nurture culture, remove organisational impediments, support leaders and teams, with autonomy, pull not push. Leadership: there to ensure high alignment, that there are clear outcomes with feedback loops and organisational guardrails to ensure that it's agile not fragile and to enable both speed & control.

# Experimentation

Now that you have a hypothesis-driven desired outcomes with measures and you have autonomy within guardrails, with coaching and support in a fractal manner, starting small, it is time for **Experimentation** with fast feedback in order to make progress on your desired outcomes.

As you are performing unique complex work within a unique complex adaptive system, you need to probe, sense and respond in that order. As with any experiment, everything is a hypothesis to test. The outcome of the experiment is not guaranteed as the system of work is emergent. Acting in the space, changes the space. Due to this, consider that many experiments may not be reversible and cannot be replicated, as the complex adaptive system will already have reacted to the input. As per Systems Thinking, there is not a linear cause and effect.

To get started, I find the following model for navigating change, from Dan North, works well:

> "*Visualise, Stabilise, Optimise*", source

**Visualise**: Before you can run experiments, first you need to 'see' and measure your current system of work. Pick a big wall and physically map out your workflow steps and the work in the system. End to end, not just in IT. For the vast majority of teams they have never before 'seen' their knowledge work like a factory floor and it is a shock. Impediments become apparent. Add aging on the tickets. Typically work is waiting 90% of the time in large organisations. This is where the waste is, slowing down the system, like too many cars on the motorway. You should Know Your Flow.

> **Look where the work isn't**

**Stabilise**: Control the amount of work in progress (WIP) in order to stabilise flow. Start putting fewer cars on the motorway. Introduce WIP limits, meaning that no work can be started until there is a free slot. The system now becomes a pull system, with work being pulled to the right.

> ***Stop starting, start finishing***

**Optimise**: Identify what you believe your biggest constraint to flow is and focus your improvement efforts on alleviating it, starting small, with fast feedback. Run one experiment (probe) and inspect the Outcome measures (sense). Remember that cause and effect are not linear. Work out what your next improvement experiment will be and go again (respond), amplifying positive experiments and dampening negative experiments. Make use of all bodies of knowledge, consider all frameworks to identify optimum practices for your unique context. Causal loop diagrams are a useful tool.

Once insights are generated from the improvement experiments, practice double loop learning to reflect back on the original Principles and Outcomes. Reflect, with contextual evidence, whether they should be updated as you gain more insight.

## Be an Omnist

Any frameworks, practices or tools are a departure point, not a destination. Be an omnist. Draw from multiple knowledge bases, don't only have one tool in the toolbox. Find your own way.

> *"Omnism affirms the necessity of one arriving at an understanding of reality based on personal experience, engagement, and inquiry, and an acceptance of the validity and legitimacy of the differing understandings of others. In this, there is, however, an implied system of values."* (source)

---

## Summary

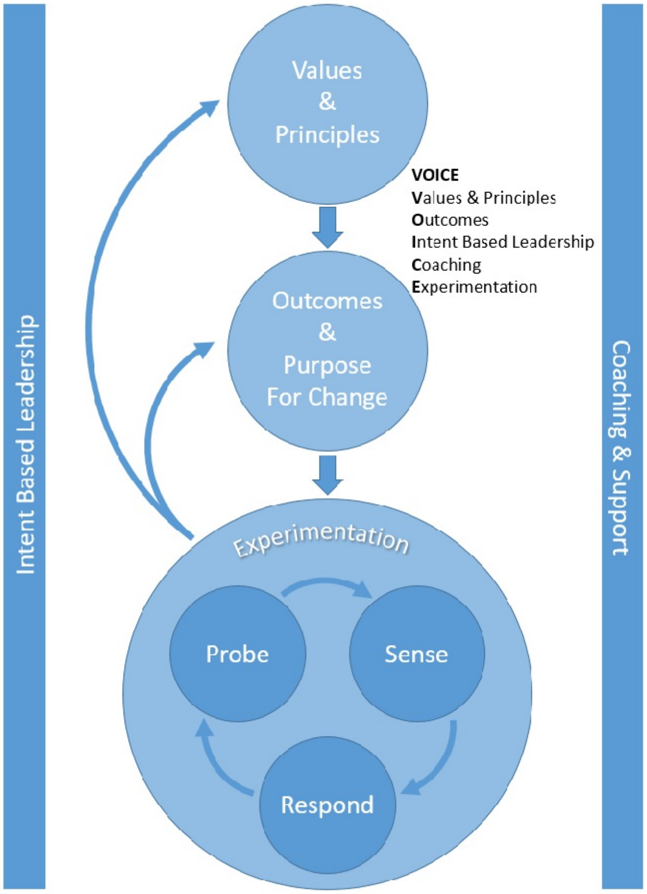
As per Drive (Daniel Pink, 2009), innate motivators are Autonomy, Purpose and Mastery.

**Values & Principles** and **Intent-based Leadership** give people *autonomy* within *guardrails*. **Outcomes** provide *purpose*. **Coaching** & **Experimentation** help people to build *master*y.

> **OH: "It doesn't feel transformational because the people affected by the changes are the ones driving those changes"**

This approach shifts the locus of control of the complex adaptive system inward, such that people are invited to be, and with safety and support become, masters of their own destiny. In some cases, this requires overcoming an entire career of conditioning that the locus of control is external, that both the how and the what are prescribed. For some people, cultures, contexts and complex adaptive systems, *Kaizen*, evolutionary change is needed first, to show that the sky won't fall in, to build confidence.

In order to deliver better business outcomes, to engage colleagues and delight customers, for change to be successful and lasting, don't Inflict Agile, instead **give people a VOICE**.



Series:

1. Want to do an Agile Transformation? Don’t. Focus on Flow, Quality, Happiness, Safety and Value
2. Want to scale agile? Don’t. Descale the work first. Achieve big through small

See also:

- In praise of SWARMing, Dan North
- Agendashift
- OpenSpace Agility

Also by the author:

- Know Your Flow
- Agile Manifesto Values add on for large enterprises
- The PMO is dead, long live the PMO
- The Yin and Yang of Speed and Control

Acknowledgements: thanks to Zsolt Berend, Mike Burrows, Tony Caink, Steve Denning, Mark Lines, Angie Main, Dan North, Simon Rohrer for feedback.

***

To read this article online with embedded hypertext links, go here:

**https://medium.com/sooner-safer-happier/organisational-agility-give-people-a-voice-5d5e68449aa7**

# About Jonathan Smart

**Jon** leads Deloitte's Enterprise Agility practice, helping organisations to deliver **Better Value Sooner Safer Happier**, through the application of agile, lean and DevOps principles and practices, organisation-wide.

Previously, Jon was leading on Ways of Working globally across Barclays Bank, which is 328 years old, with 80,000 colleagues in 40 countries. After three years, teams are on average delivering **three times as much in a third of the time with 23x fewer production incidents and the highest ever employee engagement scores**.

Jon has **25 years' experience** of taking and leading an agile approach to change. He and his team is the winner in the category of **"Best Internal Agile Team"** at the Agile Awards 2016.
Jon is the **founder of the Enterprise Agility Leaders Network**, is a board member of the SD Agile Learning Consortium, is a member of the Disciplined Agile advisory council, a member of the Business Agility Institute advisory council, a member of the Programming Committee for the DevOps Enterprise Summit, a guest speaker at London Business School and talks at 6-7 conferences a year.

# What is Agile Leadership

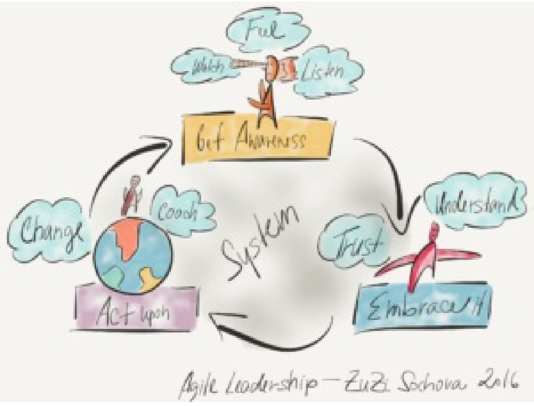
By Zuzi Sochova

Posted on April 21, 2018

What is agile leadership about? How would you define it, how would you explain it? Those are just a few questions people are asking these days.

Agile leaders are able to inspire others, creates and communicate an appealing vision or a higher purpose which motivates organization and would constantly be looking for better ways of working through feedback. It's about being inclusive, support others on their leadership journey. Being open to new ideas, experiments and innovations. Support creativity. Be able to garden the right mindset and creates cultures based on collaboration. Agile leader is a coach and good listener.

Agile leadership is not about tools, practices or methodologies. It's an ability to look at the organization from the system perspective, understand system dynamics, be able to get awareness about what's happening, embrace it, understand it and become an integral part of the system and finally be able to act upon and influence it with coaching, and initiate a change.



New management paradigm is about collaboration and trust, decentralization, continuous adaptation and flexibility, cooperation and teamwork. From the static management in the industrial era, we shifted to strategic management in last twenty years of the last century and moved quickly into the dynamic management which tries to keep up the speed with modern constantly changing, complex world. That's the world which critically needs agile leadership as anything else is not flexible enough to deal with nowadays challenges. Companies need more creativity, collaboration, and innovations – simply agility. That's the way how to be successful. That's the way how to achieve their goals. That's the way how to survive and don't become new dinosaurs who are so huge, slow and inflexible that they eventually disappear from the world.

***

To read this article online with embedded hypertext links, go here:

**https://agile-scrum.com/2018/04/21/what-is-agile-leadership/**

# Agile HR

By Zuzi Sochova

Posted on August 31, 2018

Agile HR or if you want Talent Management as it is called nowadays turn the whole company around. It's employees centric, delivering value to the whole organization. At a glance, not much had changed. We still need to hire people, take care of people growth, do some evaluations. Just the way we work changed significantly. So let's go one by one to see the shift.

### Hiring

Hiring process focuses not that much on skills, because skills could be learned, and will change depending on the business value priorities, and the team needs, but a person who is a good match to the company culture and the team. In an Agile organization people who can learn fast, are the starts. They can go to any cross-functional team and deliver value. We look for someone who has not a fixed mindset, is ready to change. Having said so, people are often not hired by HR and managers but the teams and the HR are only consulting and coaching teams in that process. The world of the fixed positions is over. All the recruiting agencies need to adapt as well. When we've been hiring, we involve team members and give them a strong voice in the process. We stopped looking for C++, Java, or C# experts, we were looking for passionate people who have energy, passionate about anything they did. We want to hear stories about what they love to do. Even if it was just a tiny thing they did over the evenings. We were transparent on how the work is going to look like, stressing the downsides, so they have clear expectations. Transparency is the key, so one of the great ideas is to invite candidates to join a team for a day. It's like going to the date, getting to know each other better, get a sense on both sides how is it going to be.

*One example of a very different interview is to ask the candidate to use a creative set of Lego bricks and visualize how it's going to be once they joined the organization and have a conversation about the model. It's something you rarely see in the interviews but it shows a lot about the candidates.*

### Evaluations

Evaluations and performance reviews changed significantly in Agile space. It's less about reviewing, performance, and evaluation, more about development and vision of the future and growth. As the Agile organization operates internally in very short cycles, where through radical transparency and instant feedback through retrospectives the organization gets to inspect and adapt and solve any issues right away, we don't really need classical KPIs as they are not supporting the adaptivity and flexibility Agile organizations need and missing a team aspect as well. As a first step, you can start with setting team goals, instead of individual ones. It will help. However, eventually, you need to redesign the whole concept from the scratch. The key focus is on coaching conversations, transparency, and candid feedback from your peers.

*One example of a radical change you can use is the team-oriented feedback. You give each person on a team or organization (yes, it scales) a certain amount of money to give away. Let say $100, and ask them to distribute it to the colleagues. The only rule is you can't keep it. If you think about it, the message you got by receiving $0 it's much stronger feedback then anything your manager can ever say about your performance. Indeed, we need a lot of coaching to help people understand and handle what's going on, but in general, that's a good thing. If you scale this to the whole organization it's even more fun, as the managers get such instant feedback as well as the employees.*

### Talent management

As I mentioned at the beginning of this article we are speaking more about talent development then HR. What motivates people? How do we grow talents? How do we support them on their journey? How can we help them to be successful? The answer is coaching, support them to create their own development goals, grow their interest, empower them, raise their awareness about themselves. Not surprising, but how many HR are taking such a support role and how many of the companies take it as process and governance role.

*Example of such coaching conversation for the people growth could be using a few categories which are strategic for the organization right now to frame the conversation. Firstly, you need to make people aware of how the coaching scale works, that it’s very different from evaluation, it doesn’t have to grow quarter to quarter and that there is always a better way of doing things, and that this tool shall help them to identify their potential and find ways how they can grow to support the organization. As a next step, you let people rate themselves on a relative scale 1..10, where 1 = not good at this area, and 10  = I’m great at this. They need to be able to compare themselves with the other people around in the organization, explain how it would be, when you are 2 points above the level you are currently, what would be different once you get there, what would it mean to the organization, what is currently in their way, etc. All of those are good coaching questions. No magic. It just works like a magic* ☺

***

To read this article online with embedded hypertext links, go here:

**https://agile-scrum.com/2018/08/31/agile-hr/**

# Agile at executive team level

By Zuzi Sochova

Posted on September 27, 2018

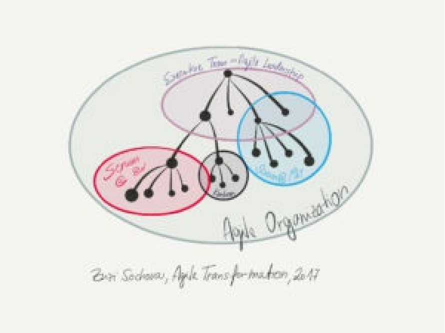
Agile can't stay just at the team level. Agile transformation only creates disturbance and gap between the management and employees. And the more Agile the teams are, the bigger the disconnect is. Managers feel lost, forgotten and start to be frustrated that those self-organizing teams might eventually not need them. Part of the problem is they've never been part of any Agile or Scrum team themselves.  They've seen them working, joining them for Reviews and listening to their stories, but that's not the same. People need experience to understand a different way of work. You might still remember your first feeling, when someone told you that this Agile and Scrum will be great. "What?" you thought, this stupid process will never work – what if… At least I still remember how I felt several years back.

Leaders & Teams dimension
Agile Transformation, Zuzi Sochova, 2018

One important thing companies often forgot during their Agile transformation is how to get management on board. Managers deeply need their own experience with Agile and Scrum. They can't just read about it. Otherwise you continue hear such funny remarks like "I got it, you are a team, you collaborate, but who is responsible?", "We don't need ScrumMasters, some developer can take it as a second role" or "We don't need Product Owner, we have a product committee". If you really mean the Agile transformation seriously, it's time to change the way you implement it. It's not just a different process decided by C-level executives and implemented without them noticing. It's a significant change of the culture and mindset. So why don't we start from the other side, forming the first team from executives. Let them experience Agile and Scrum. Make them feel the pain of being the group of individuals with their own goals, no common passion, no trust. Or no unifying purpose. Let them experience what the self-organization is about, how the cross-functionality works. Let them do their refinement, planning, standups, reviews, and retrospectives. It's always fun. And the same way as such first pilot is painful and difficult for a product development team, it is even more painful for the executives. They would hate it. If they can, they would kick you out of the door. So be ready for that and have strong enough sponsor who understands that such painful experience is critical for the organizational success. It's like any other exercise. Starting is difficult. We all are great at finding excuses why running today is just not a good idea. I will run tomorrow. Or when it's the right weather. Actually, I don't think I need to run, I'm just good without it. The other people need that, not me. Familiar? If you force yourself to start and develop a habit, it is fun and you would miss it if you skip that for even a day. The same with Scrum, the first time you experience the power of the true team spirit you never want to be back. No matter where you are in the company orgchart. It works the same way.

Unfortunately, executives are rarely going that way. There are two reasons. First, it is a painful journey. That's why I'm not running every day. It's not that bad that I would have to, right. The company is still fine. Not struggling enough. But maybe when that happens it's too late to change. Second, most of the Agile Coaches need a day job. They care about 6+ months contracts. They are afraid of losing it if they would push too much. They often forgot that their job is not to please the customer, but to change them. Guide them through that painful experience with all the risks that they will not like it, and stop. Very often you hear from them "I know that this is not the way how it shall be but this is a corporation, you have to do it differently" so they still have PMOs, no Product Owners, weak

ScrumMasters and not real teams either. It’s a much more painful experience for everyone involved then starting this fake transformation repeatedly all over again and again.


Executive Team = Agile Leadership
Scrum @ Biz
Agile Organization
Zuzi Sochova, Agile Transformation, 2017

If you mean it, get a real Agile coach. Not a consultant. Find someone who would guide you how to do it. Not do it instead of you. Who would be with you once per Sprint / month / quarter. Do their intervention, show you the way where to focus next and let you exercise. Start with smaller pilots. The first Agile bubbles. The more bubbles you make, the better. Aim for our own experience and learning. Inspect and adapt. Don’t forget that your executive team forms one of the first bubbles. They have to learn in the first wave. If you do it that way, Agile mindset will grow organically and very soon you would be ready to share your own Agile success story and inspire the others.

***

To read this article online with embedded hypertext links, go here:

**https://agile-scrum.com/2018/09/27/agile-at-executive-team-level/**

# About Zuzana (Zuzi) Sochova

**Zuzana** “Zuzi” Šochová is an independent Agile coach and trainer and a Certified Scrum Trainer (CST) with more than fifteen years of experience in the IT industry. She started with agile and Scrum back in 2005, when she was implementing agile methods in the USA. From that time, she has been credited with agile transformation and implementation for many companies and teams around the world. By creating and sustaining agile leadership, Zuzi believes the worlds of work and life can be made happier and more successful.

Zuzi is a founder of the Czech Agile Association, and the world famous Agile Prague conference.

Zuzi was elected to the Scrum Alliance Board of Directors in 2017. She is an author of *The Great Scrum Master: #ScrumMasterWay book (Addison-Wesley Signature Series (Cohn)*).

Her website is sochova.com and blogsite is agile-scrum.com.

***

# Remote Teams and Virtual Facilitation

By Ram Srinivasan

Posted on October 10, 2018

I have my own biases,  having seen co-located teams work very well and having struggled with “dislocated” teams, I often recommend my clients to reconsider “dislocated” teams.  And by “dislocation” I am not saying you cannot have teams ten time zones away.  What I mean by a “dislocated” team is when three team members are in one time zone, but in different locations and other team members are in different locations and time zones. (Technically speaking – it might be better to call them workgroup than a team).  A “non-dislocated” team is one where all team members are physically sitting right next to each other.

Research (here, here, and here) has shown us that the the cost of “dislocation” is way too high than the cost of “co-locating” the team. Often times, the team members are dislocated from wrong reasons (and I won’t go into that here). My suggestions usually get brushed off because (a) the clients have far more important problems to deal with (b) the decision maker is way higher up in the totem pole and my message does not reach them or (c) clients would rather run behind a tangible cost saving (e.g. cost)  than looking at the intangible loss (e.g. loss of high bandwidth communication between dev. team members due to time zone and technology challenges which ultimately increase the cycle time of features).

One of my friends challenged me to attend a workshop last Thursday on “remote facilitation”. I know many of my students and clients struggle with this and  I accepted the challenge. The content of the workshop blew me  away (**Note: – colocated teams are still better than dislocated teams, I have not changed my mind on this. I am merely sharing a few ideas that I learned so that you can find them beneficial**).  Here are my few major take-aways.

- Facilitator tip – Rather than merely trying to replicate a technique that works for  in-person meetings, try to de-construct why that technique works and reconstruct that technique for virtual meetings
- Participate in the virtual meeting with the same level of attention (or more) and engagement as though it is an in-person meeting (that means no multi-tasking)
- There is a self-fulfilling prophecy with regard to virtual meetings  – you experience poor virtual meetings, you expect bad meetings, you get bad meetings, and the cycle perpetuates itself
- **Set expectations upfront** – very clearly and this is how you break the self-fulfilling prophecy
  - Ban phone only meetings, use videos for ALL the meetings.
    - People get a lot of cues when seeing face and having a video helps in non-verbal communication, not to mention that it actually engages people
    - Have you heard the toilet flushing sound when in conference calls(because someone forgot to mute their phone)? Bet you will not hear that when you have your participants turn on the video.
  - You got to be on the video, else we close the meeting right away. You join on video, else you don’t. Pe-

    riod.
    - You got to join the meeting from a quiet place, not "dial-in" it from the bus when on you are on your way home . And you must be on the video. Period.
    - Even if one person breaks the expectations once, we close the meeting right away. We break it once, it is an excuse to break it the second time and we are back with the self-fulfilling prophecy of bad virtual meetings
- Normalize the communication channels – One person is remote? Then everyone is joining remotely using their own video from their laptop.  Two people cannot join using the same video. Don't have a camera? GET ONE !!
- Facilitator and participant tip – try having the video right below the camera (than having the video in different screen) in your laptops/computer. It creates an impression that you are looking into the camera when you are looking into the video
- In-person meetings and co-located teams work because we "socialize" quite a bit. Try having some "social" time in virtual meetings as well. Try "bring your own cider" (the choice of drink will depend on the timezone of the participants)
- As a facilitator, you got to have everyone engaged – here are a few tips
    - Get everyone on video.
        - This minimizes the participants' tendency to multi-task
        - This also prevents people from anonymously snooping in. Have you had people join a conference call and not announce themselves?  Will you let someone walk into your in-person meeting with a mask on? If no, why would you have someone snoop into your virtual meeting?
    - Avoid PowerPoints – it is just one way broadcast. Use tools that support "virtual" break-out rooms.
    - Increase psychological safety (more on this in a different blog later) so that people can actually speak up.
- Facilitator tips –
    - Like my friend Mike Dwyer says – use the NOSTUESO rule – No One Speaks Twice Until Everyone Speaks Once.  And the participant has the right to pass.  This creates space for people to speak up.  Also, if participants speak up in the first five minutes, they are much more likely to speak again.
    - Hard to pass a talking stick and figuring out who should talk next in a virtual meeting when facilitating round robin discussion – try this idea –  Have a participant speak and then nominate the next person. And repeat till everyone speaks
    - Prepare… prepare… prepare. You cannot wing a virtual meeting. You need more preparation. And you need a Plan B as well. What if the internet connection fails? What if your laptop crashes?
    - Pay attention to discomfort – participants can only sit in once place for so long
    - Bring psychological safety and engagement from everyone into the working agreement.  What might be the few ways that we damage psychological safety (sometimes unconsciously)?
    - Have someone paraphrase what a speaker said. This makes people pay more attention and also ensures that the speaker's message landed as intended
    - If appropriate, use tools like https://www.mentimeter.com/ or https://kahoot.com/ to increase engagement during the meeting by having participants answer questions.
- When women speak first, the probability that other women speak is higher.
- Remote meetings are a lot smaller than in-person meetings. It is hard to have more than 12 people in a virtual meeting (and then expect them to be engaged). If you are new start with six, then build up.

I hear what you are thinking, "But… But … But…. in our office, we cannot do…..".  If, you expect different results, you need to change. Because someone once said, "Insanity is doing the same thing over and over again and expecting different results."

My question to you is, "**What is possible for you to change (and keep improving) to get better results?"**

What other tips do you have?  Please share it here

***

To read this article online with embedded hypertext links, go here:

**https://innovagility.com/2018/10/10/remote-teams-and-virtual-facilitation/**

# About Ram Srinivasan

**Ram** Srinivasan has a very unique distinction of being the first person in North America (and the second person in the world) to hold the coveted Professional Scrum Trainer (from scrum.org) and Certified Scrum Trainer (from Scrum Alliance) certifications. His mission is to help his clients build great organizations and he does this by focusing on people, process, and product development.

Ram started as a developer and later moved to hold various roles including software architect, Project /Program Manager, and Agile Coach. He has coached, mentored, and consulted with teams to organizations of different sizes across many industries (finance, insurance, retail, banking, media, telecom, government, etc.). This depth and breadth of experience give Ram the ability to view clients' challenges from different perspectives and provide pragmatic solutions, making him a sought after Agile Coach.

Beyond Agile and Scrum, Ram’s interests are in Emotional and Social Intelligence (for individuals, groups), Applied neuroscience for organizational and culture change, System Thinking, Complex Adaptive Systems, and understanding Human Systems in organizations. He is a frequent speaker at Agile meetups and conferences. From time to time, Ram also runs free coaching circles and book clubs.

Ram holds a Masters degree in Engineering from The University of Arizona and is currently based out Boston, MA. You can reach him at Ram at InnovAgility dot com or follow him in his blog at https://InnovAgility.com

# How Agile helps non-technical teams get things done

## Here's how government workers at the City of San Jose adopted Agile methods to boost efficiency and productivity.

By Michelle Thong

Posted on August 15, 2018

**Image by :** *Thomas Hawk on Flickr. CC BY-NC 2.0. Modified by Opensource.com.*

What are the best ways for governments to improve effectiveness and efficiency? At San Jose City Hall, we're getting traction with an unconventional approach: agile for non-technical teams. Public servants who do everything from emergency management to parks programs are finding that Agile methods help them with that most basic of challenges: Getting things done amid frequent interruptions and evolving priorities.

Last September, I proclaimed, "Scrum is the best thing that's happened to our government team." Our innovation team of five had discovered that planning and delivering work in small increments enables us to stay focused, aligned, and continuously improving. We didn't yet know if our experience would be replicable by other teams in our organization. We offered Agile training for 10 colleagues to see what would happen.

Nine months later, 12 teams and more than 100 staff members throughout our organization are using Agile methods to organize their work. Notably, the spread of Agile among city teams has been largely organic, not driven by top-down mandates.

## What is Agile?

Agile is a way of organizing teams to work in an iterative, incremental, and highly collaborative manner. There are many different forms of Agile, but two of the most common are Kanban and Scrum. While Kanban is a lightweight framework that focuses on visualizing tasks on a board (physical or digital), Scrum is more prescriptive and complex, with specific roles, meetings, and time-boxed sprints. Both Kanban and Scrum place value in responding to change over following a plan, continuously improving process, and giving the team autonomy to determine how the work gets done.

While there are many variations on Agile methods, here are some aspects that are common among our teams in the City of San Jose:

- The team maintains a list of all potential tasks, known as a "backlog"
- Tasks are not worked on until explicitly prioritized
- Work is made visible by tracking tasks on public boards

- Team conducts regular check-in meetings
- Team conducts regular "retrospective" meetings to identify on what’s working well and what needs improvement

While Agile originally developed in the manufacturing industry and became well-known as a software development framework, it has been adapted to other sectors, including government.

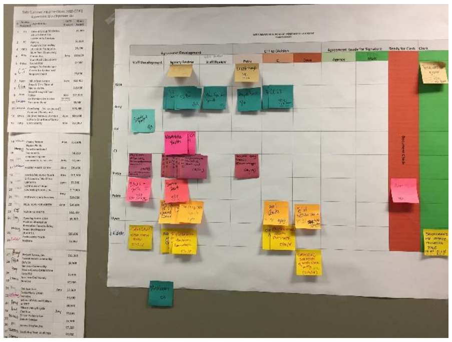
## Why city leaders are choosing Agile

Of the dozen teams using Agile in the City of San Jose, almost none are delivering software, and almost none had heard of Agile a year ago.

Why did these teams decide to go Agile? Here are the top reasons we hear from internal Agile champions.

**Aligning as a team.** When Ramses Madou began managing the City of San Jose’s transportation planning group, he saw a need to align the contributions of his talented team members into a coherent effort that advanced the department’s overall priorities. Before implementing Scrum, he says, "Everyone was running on their own track."

**Focus on priorities**. After the launch of a new grant program, San Jose’s Parks, Recreation and Neighborhood Services department suddenly found itself with 47 contracts to process in six weeks. To help the team get a handle on the work, program manager CJ Ryan enlisted our resident Agile coach Alvina Nishimoto to guide the team in setting up a Kanban board to visually track the contracts. After two weeks, CJ could see that the team was not progressing as quickly as needed, so she decided she needed a 100% focus. She blocked off her team’s calendar for two full days and gave them air cover to ignore other demands on their time. Without constant interruptions, her team got through 25 drafts contracts in just two days.



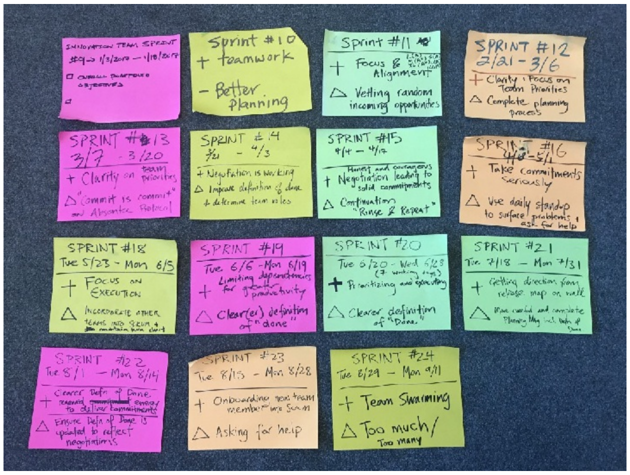
Team Kanban board for tracking contracts (Photo credit: Alvina Nishimoto), (CC BY-NC-ND 3.0)

**Manage interruptions.** The web team for San Jose Public Library receives a constant flow of requests from other library divisions, which makes it difficult for the team to make progress on important longer-term projects. When the team implemented Scrum, the team’s manager and "product owner" Laurie Willis became the filter for all requests, ensuring that they would be worked on only after being prioritized for a given sprint. Team member Stacy Tomaszewski immediately started feeling the benefit of not being pressured to respond immediately: "We feel like we can say no." And in government, where everything feels like a priority, the ability to say "wait until our next sprint" can be very powerful indeed.

Daily stand-up meetings allow team members to share progress. (Photo credit: Andrea Truong), (CC BY-NC-ND 3.0)

## 5 tips for getting started

**Start with the basics: Make work visible**. Transparency is one of the first benefits experienced by most teams in our organization. To get started, all you really need is to put prioritized tasks up in one place where you can progress from "backlog" to "work in progress" to "done." Team members benefit from having a visual reminder of their priorities and getting to share their progress with the team. Managers and stakeholders get more visibility into work underway. We recommend starting with a simple, lightweight system. If your team is co-located, it's hard to beat a physical board in terms of a visual reminder. Most San Jose teams still use sticky notes to track tasks. A few teams use Trello or other digital tools and have successfully gotten team members in the habit of checking and updating Trello regularly.


INNOVATION TEAM SPRINT #9 → 1/3/2017 – 1/18/2017
□ OVERALL PORTFOLIO OBJECTIVES
Sprint #10
+ teamwork
– Better planning
Sprint #11
+ Focus & Alignment
△ Vetting random incoming opportunities
SPRINT #12 2/21 - 3/6
+ Clarity : Focus on Team Priorities
△ Complete planning process
SPRINT #13 3/7 - 3/20
+ Clarity on team priorities
△ "Commit is commit" and Absentee Protocol
SPRINT #14 3/21 - 4/3
+ Negotiation is working
△ Improve definition of done & determine team roles
SPRINT #15 4/4 – 4/17
+ Honest and courageous Negotiation leading to solid commitments
△ Continuation "Rinse & Repeat"
SPRINT #16 4/18 - 5/1
+ Take commitments seriously
△ Use daily standup to surface problems + ask for help
SPRINT #18 Tue 5/23 - Mon 6/5
+ FOCUS ON EXECUTION
△ INCORPORATE OTHER TEAMS INTO SCRUM + maintain burn chart
SPRINT #19 Tue 6/6 - Mon 6/19
+ Limiting dependencies for greater productivity
△ Clear(er) definition of "done"
SPRINT #20 Tue 6/20 - Wed 6/28 (7 working days)
+ Prioritizing and executing
△ Clearer definition of "Done."
SPRINT #21 Tue 7/18 – Mon 7/31
+ Getting direction from release map on wall
△ More careful and complete Planning Mtg incl. Defn of Done
SPRINT #22 Tue 8/1 – Mon 8/14
+ Clearer Defn of Done renewed energy to deliver commitments
△ Ensure Defn of Done is updated to reflect negotiations
SPRINT #23 Tue 8/15 - Mon 8/28
+ Onboarding new team member into Scrum
△ Asking for help
SPRINT #24 Tue 8/29 – Mon 9/11
+ Team Swarming
△ Too much/ Too many

A record of "pluses" and "deltas" shows what worked and what to be improved from 15 sprints over 8 months. (Photo credit: Michelle Thong), (CC BY-NC-ND 3.0)

**Iterate and improve over time**. Continuous, incremental improvement is the real secret to the power of Agile. All of San Jose's teams do regular retrospectives to reflect on what's working and what needs to be improved. This process empowers your team to make Agile work for them and overcomes resistance by allowing the team to drive small experiments that add up over time. CJ Ryan's team in the Parks department initially resisted the idea of daily stand-ups, and she decided not to push them. Instead, the team started with a simple Kanban board. After a while, the team decided that daily stand-up meetings would be useful so they could share progress at a finer-grained level. Next, they decided to start regular planning meetings as a team to establish priorities. Effectively, the team has evolved to adopt many "Scrum-like" practices, but at a pace that makes sense to them.

**Get some expert help to take it to the next level.** While it's easy enough to start adopting Agile practices you read about online, troubleshooting can be hard if your peers and team don't have prior experience with Agile in a range of contexts. We've found it critical to have the wisdom of an experienced "Agile coach" to support teams, provide guidance, and offer spot training and coaching. We found both our coaches through the Encore fellow program, but you could also hire an Agile consultant or perhaps find pro bono volunteers from the private sector.

**Be patient.** Agile helps teams to establish productive habits like prioritizing, asking for help, and reflecting. But Agile is not a magic bullet—it won’t solve all your problems overnight. In fact, it may even make some of your problems more apparent. Give yourself time to figure out how to make the process work for your team.

**And finally, have fun with it.** A big benefit of Agile is that it builds connections among team members, enabling deeper collaboration and greater effectiveness as a team. So don’t be too serious about it. Instead, experiment with creative ways to make Agile an enjoyable and integral part of your team culture.

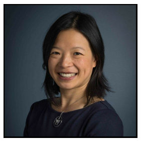
Let us know what you learn along the way.

***

To read this article online with embedded hypertext links, go here:

**https://opensource.com/article/18/8/agile-helps-non-technical-teams**

# About Michelle Thong



**Michelle** Thong is a public servant, product leader, and service designer. In her role as Digital Services Lead for the City of San Jose, she is building the team and strategy to deliver user-centered government services to over 1 million residents and 60,000 businesses.

Michelle has previously worked as a circuit designer, project manager, and urban economics consultant. She holds Bachelor's degrees in Electrical Engineering and Feminist Studies from Stanford University, and a Master's in Urban Planning from the University of California Berkeley.

# The Flying and The Thud: Mental Health Issues on Agile Teams

By Cass Van Gelder

Posted on August 1, 2018

## About This Publication

Over my ten plus years of working in Agile, I’ve seen a preponderance of ADHD, OCD, anxiety, Asperger’s Syndrome, autism, Bipolar Disorder I and II, etc. Additionally, since Agile encourages teams to work together, these issues can be a stumbling block—and sometimes a 50-foot wall—that keeps the team from growing, and even from simply functioning. While supervisors have supported ridding the teams of these “difficult” team members, I have advocated maintaining the members and tweaking our approach to be more inclusive.

## 1. INTRODUCTION

I’m writing this in the middle of a tornado of emotion, information, and desire to know more. By the time this is published – a few months from now – I expect even more will have changed.

Earlier this year, very prominent people – Kate Spade and Anthony Bourdain – killed themselves, and their family members attribute those deaths to the mental issues both people experienced. Ms. Spade’s family recounted how she was afraid to seek treatment for fear of how she would be viewed, resulting in a potential financial loss for her company. Mr. Bourdain’s family described how he sought treatment, but then promptly ignored it.

In our own software development community, we hear stories similar to Joseph Thomas’ [1]; a man who excelled prior to working in Uber’s “intense work culture,” committing suicide a mere 6 months into his employment. Prior to Uber, “…this (was) a man who didn’t have a history of crumbling under stress,” Richard Richardson, the family’s lawyer, said.

The windows are slowly opening, whispers of darkened smoke eek under the doors. The conversations, much needed and important, are beginning.

With these events in mind, I move forward with what I already believe to be an extremely important topic – mental health issues on Agile teams.

## 2. It’s Not Them, It’s Me

I come to this topic with more knowledge than I’d care to have, having lived the majority of my life without vital information. From the time I was old enough to remember, life with my mother was either a whirlwind of thought and extreme activity or it was completely devoid of movement and motion – The Flying and the Thud, I called it.

The Flying was at times magical, filled with creativity and excitement, a whirl of pixie dust surrounding everyone near it. My mother’s trip to Wal-Mart when we were young involved us watching her load the cart with everything she could remember from an imagined shopping list. She’d picked up a hammer, nails, a vent cover, paint, then she’d see a sewing kit out of the corner of her eye. Her mind flitted to all the holes and buttons in our clothes that needed repairing. She began grabbing needles, thread, seam rippers from the display walls; snatch pre-cut fabric, start talking about the quilts she would make us from the scraps, then remember she had wood leftovers in the house that needed to be used and wouldn’t a new wood burning stove be perfect, resulting in another aisle grab of more nails, a book on building your own fireplace.

As we circled the store, sometimes for hours on end, our cart would quickly fill and eventually, she would begin to tire. The Thud was coming on, her eventual fall to the ground that made her forget this world ever existed.

By the time we'd finally reached the checkout counter, she'd either rid the cart of everything except contact paper, unripened bananas, and hair clips, or she'd buy everything, barely having the energy to remove anything from the cart, and even less to bring it all into the house. She'd limp into her bedroom, closing the door, sleeping in her dark room for a solid 10 hours. The small piles of her purchases collected dust for months on end, until finally, mercifully, they were put out in the front yard for a garage sale.

Having grown up in that environment, I did not know anyone lived differently. This was the normal I expected everyone experienced.

It was not the first time I was wrong.

I had my own flying and thud. In my own head, waves came crashing at me hard for a random block of three days. I would feel an exceptional rise in my creative thoughts. I would spin from one topic to the next, interrupting anyone speaking because I could think through at least five possible endings to what they were about to say, and had already concocted a response. I would forego sleep so that I could play piano until 2 am, or, trance-like, write a 200-page story, or design an entire set piece for an imagined play, complete with working lights. These waves resulted in orchestral symphonies, complete plays, paintings, drawings – all of them done while, what I referred to as my secondary brain, was occupied by books on tape or PBS documentaries. When I returned to look at the drawings, in particular, I could recall every detail of the documentaries I had listened to as though they were in front of me again. This was my flying.

These bright spinning colored days were always followed my own thud, the ungraceful drop to the ground that would break my emotional bones. While they could be triggered by a horrific news story, a gut-wrenchingly miserable song, or an emotionally encompassing film, usually the thud stumbled in on its own, pulling me down and dyeing everything with a wash of dark grey. It was rare that the thud resulted in tears; instead, I continued on, feeling as though the string had been cut from the balloon that kept me tethered to this world. I felt no connection to it and no desire to continue it.

School allowed me to hide my issues, my inability some days to put full focus to anything, to flit around the room with thoughts and tangents. I easily finished full assignments the period before it was due, all the while absorbing the current class's lesson.

As grown-up, work was harder. Unlike others who suffered as I did, I did not subconsciously self-medicate. I had no taste for alcohol or drugs. I suffered the flying and thud at full force, no buffer.

Unlike school, co-workers and supervisors were annoyed by my fits of creativity as they came packaged with my irritation, bluntness, and ego. My sick days flittered away easily when I could not force myself to function. After an unexpected breakup in my personal life, I spiraled into a thud that left a deep imprint on the Earth. For days, I sat crying even though I was holding perfectly normal conversations. A concerned co-worker/friend drove me to the hospital, having to explain to me that how I was reacting wasn't typical.

The majority of the time, it felt as though I had been secretly stuck with a needle of potent drugs, over and over, with no control over what would happen.

At one of my previous employers, my then-boss wrote me up for those same indicators. My inability to focus during meetings, my interrupting, my absences, my tardiness, my distractibility – had infuriated and baffled him. His requirements for my continued employment were strict: I had to forego my cell phone completely, control my interruptions, only be absent for pre-approved vacations, not miss nor be late for any meetings, and be completely focused during hour+ long meetings. I would have to continue this under his watchful eye for six months, if I was to remain at the company.

The first week of these requirements I felt as if heavy chains had been attached to me as I moved. I was without improvised tools that kept me focused during droning meetings. There was a real possibility to me failing spectacularly, and I knew this company had no problem ridding themselves of difficult people.

The prior year, Carl (not his real name) was my co-worker who I saw struggle with similar issues but on a grander scale. I had coached him several times prior to the restrictions and after to help him find ways to be successful. I watched him struggle as if in physical agony while trying to meet his boss's expectations. It was not the way he worked, Carl told me. He felt as if he were being put into a box, then forced to walk five miles while twisted inside. When he was given similar restrictions and failed to meet them within the sprint, his boss fired him.

Another team member, Jack (not his real name,) struggled later the same year when he was written up for the third time for the same infractions. As a Scrum Master, I experienced Jack's lack of focus, watching him follow many rabbits down several holes simultaneously, and distracted by people simply walking by his desk.

Jack heard every conversation within a five pod radius, and desperately wanting to be folded into an accepting group, would interject himself into discussions, interrupting and correcting, and easily irritating others. After being written up the first two times, he questioned not only the individual items that had been listed in his downfall, but escalated the issue, causing even more contentiousness. The third time was the charm, however, and they let him go before the end of the last restrictions expired.

My determination was simple: I would not fail – and, I did not. However, in doing so, it meant opening myself up to the possibility of causes, causes I did not want to admit. It meant possibly being labeled what I had for years what I had denied.

I finally decided to have myself tested.

Prior to this point, even though I had a grandmother and mother with undiagnosed Bipolar II Disorder, one child with an officially diagnosis of the same, a sister who exhibits Bipolar II traits, I still had convinced myself that it had skipped a generation.

The test took over 6 hours and, before I even left, the doctor told me what he suspected – a fact that I have been hesitant to talk about for the same reason anyone reading this might be hesitant if they have experienced the same diagnosis; a fact I struggled with even as a child, but had no name, no medicine, no exercises, no way to anticipate or to control the way my mind twisted and turned. Like an internal roller coaster, it jotted, stopped, skipped ahead, interrupted itself, curved, and left behind any original thought just to chase a new one that appear like a red balloon.

So, now, I was this thing.

I was Bipolar II.

## 3. Let's Talk Some Facts

As a whole, we'd like to think mental health issues aren't overwhelming prevalent, but according to The Center for Workplace Mental Health, 1 in 5 adults will experience a diagnosable mental illness in any given year [2]. Twenty percent of us. And that's across all sorts of fields, not just software development.

Half of those, though, will go untreated. They will suffer in silence, but they will still suffer. On top of that, many workers feel they cannot be away from work to treat their mental health issues. So, while they are physically present at work, their depression and/or other symptoms keep them from being able to function fully – a term coined "presenteeism."

According to the U.S. Surgeon General's Report on Mental Health from 1999, the costs of untreated mental health disorders results in $79 billion annual loss to businesses due to lose of productivity and absenteeism [3]. That was nearly 20 years ago.

According to Kessler et al., 2006, mood disorders are estimated to cost more than $50 billion dollars a year in lost productivity and result in 321.2 million lost work days [4].

In 2008, an abstract presented by de Graaf, et al., screened over 7000 workers in 10 countries for attention deficit/hyperactivity disorder (ADHD) symptoms [5]. Out of all these workers in a multitude of industries, 3.5% of workers showed symptoms of ADHD. In spite of this, only a "small minority" ever received treatment.

Compounding this situation, these workers were:

- Less likely to graduate from high school or college
- On average made 20 to 40% less than coworkers with no symptoms but equal education and training
- However, when employees, specifically with ADHD symptoms, DID seek treatment, 80% reported improvements in job satisfaction and productivity, an astounding result

## 4. I Don’t See the Relation

Over my ten plus years of working in Agile, I’ve seen a preponderance of ADHD, OCD, anxiety, etc. issues on Agile teams. Instead of receiving adjustments that might help us flourish, we are met with admonishments and whispers behind our backs. Managers write us up for infractions that are more symptoms than real problems. People around us pick at what they can understand and discard what they can’t.

When Carl was promoted because of extensive creative project work he completed, his once-teammates-now-subordinates reacted with passive-aggressive behavior: ignoring work directions, excluding him from important meetings, imitating him when he was out of the room. They struggled to work with his approach as a teammate. As a supervisor, they collectively decided it would be impossible.

On one particular occasion, Carl arranged a Blue Sky meeting so that the team could investigate new ways to improve the team overall (as in, if all things are possible in the blue sky, what would you do?) As they all sat in the room, the silence from the team was thunderous. No one would speak or contribute. They didn’t understand him, and he didn’t understand why.

Within a week, Carl was demoted.

As I mentioned earlier, Jack struggled in his relationships with co-workers and his boss, wherein his symptoms worsened when they rejected him. When this occurred, his boss’s and teammates’ negative reactions intensified. Because of their lack of understanding and Jack’s inability to inherently “be like them,” he was ostracized.

From that point, subjective confirmation came into play, where anything positive he did was discarded but anything negative was given double weight.

Funnily enough, by rejecting people like me with mental issues, others miss the incredible caverns of hidden surprises that come from inside our brains.

Years prior while working for a college with a registering class of 24,000+, I was put in charge of collating and distributing all the semester’s transcripts. Because the mailing labels were printed in a different facility than the transcripts, they arrived unmatched and in varying order. On top of that, we employed 190 volunteers to complete the work. As a result, it was common each semester to have approximately 3,900 transcripts that were mismatched and returned.

The years I supervised the process and volunteers, I understood instinctively how to match the labels to transcripts and how to translate enough information to the volunteers to make it easy to understand. My mind bent around the problems, flexing itself against the edges, and discovering the worm holes that caused issues. My first semester, the number of returned and mismatched transcripts: 4.

Because we are not typical, does not mean we are not worth the effort.

### 4.1 An aside

I’ll stop here for a moment to clarify something: Having Bipolar II Disorder, and having raised several children: two with ADD, one of those with social anxiety; another who is bi-polar; another who has Asperger’s Syndrome, I avoid using the word “normal.” Instead, in our house, we use the word “typical.”

Normal refers more to a state of being, as though there is only one form of human being we should all be. Anything outside those boundaries is abnormal, including over-performing people. Typical refers to what we are as a median.

It captures the overall sense of a thing. For instance, in school you might say, “Typically, at this age, six-year-olds can read most Dr. Seuss books.”

Typical is important because we all are not identical – none of us. Therefore, those of us who do not match what is typically observed are not misfit toys, ready to be discarded at the first indication that we do not mimic everyone else’s behavior.

Though we may see patterns that typically occur in each other, we are not standard, as though created from a machine. We do not move in identical ways. We do not voice ourselves as duplications of each other. We do not think the same way. And, this is to our ever-evolving advantage.

## 5. But why Agile?

On an Agile team of 5 or more people, research shows that frequently at least one (1) person is currently experiencing mental illness. The majority of those are dealing with depression, whether brought on by genetics or a situation (PTSD, trauma, etc.). So, if we had three (3) hypothetical teams of five (5) people (total of 15 people,) the likelihood would be that three (3) of them would be experiencing a mental health issue. Only one (1) of them would seek help [6].

Software development employs Agile. In turn, software developers show a preponderance of ADHD, OCD, Asperger’s Syndrome, autism, Bipolar Disorders, etc.

Additionally, since Agile encourages teams to work together, these issues can create stumbling blocks (and sometimes a 50-foot walls) that keep teams from growing, and even from simply functioning.

Two years ago during a conference in Nashville, I participated in a Kanban workshop. From the back of the room, a hand shot up during the Q & A portion. The gentleman rose and spoke of a team member with whom he and others struggled. He described how the team member acted: distrustful (his word,) interrupting, single-focused, balked at last minute changes. The list went on and on. He heaved a great, frustrated sigh as he finished.

The presenter did not. He simply said, “Get rid of him.”

*Get rid of him*.

A harsh, but likely, typical response; those of us who have these conditions face a hostile/difficult environment. Getting rid of someone shouldn’t be our first solution.

I am not a doctor. I don’t even play one on TV, so I can’t pretend that I have the medical background to be able to diagnose a person simply from hearing their story as told by yet another person. In my opinion (based on my personal experiences,) the team member likely needed some adjustments or modifications, even if he didn’t realize it.

I told this story later to my colleagues, trying to elicit their ideas about what they would suggest as a solution. Interestingly enough, many people responded that it would depend on how good of a worker she or he was as to whether or not it was worth the effort.

Let me put this into another context: if you had someone who was born needing a wheelchair, would you only adjust or make modifications for this person in the office if she or he were exceptional? If so, where do you draw the line where they would no longer be worth it?

Many people with mental issues generally are born with them, while some people have issues as the result combat, abuse, and other trauma. Many suffer from something that they were simply born with and likely might not know they have. If we only make adjustments for those of us who are exceptional, then we’re helping to perpetuate lifelong issues with under-employment, ongoing stigma, lack of proper care, and loss of emotional support.

Remember, you hired this person for a reason. Likely, that reason is a good enough one to make an effort.

## 6. Why I Discussed My Mental Health Issues at Work

For me, hiding my diagnosis – even before it was official – was a huge burden. Every moment, I was afraid I would be discovered. I spent an enormous amount of time worried I'd be found out, and then doubly worried what the consequences would be.

Deciding to finally talk about it with my colleagues was a very personal decision. Because of the stigma that still attaches itself to mental health issues, you have to judge for yourself what your true risks are.

In many ways, discussing my mental health issues at work benefitted both me and my teams. My teammates acknowledged that there was a greater sense of relief in understanding the reasons behind some of my actions. In some cases, co-workers have acted as "bumpers" for me, in that they are able to see symptoms before I can. Additionally, because we're having these important conversations, there are more adjustments that are easily made, rather than begrudgingly. When adjustments are made, for me, it can help shorten a cycle because I'm not simultaneously trying to process through a cycle, complete my work, and navigate unrealistic emotional expectations.

Having people I work with understand that there are times I don't choose to act the way I do or that I need some support is a tremendous burden lift. Additionally, there were times I wasn't unaware of my behavior. Because I had eventually elected to self-manage my mental health issues with exercises, meditation, and other natural remedies, being able to identify a tipping point helped thwart episodes immensely.

Additionally, being open about my condition makes it easier for others to discuss issues they may experience, too.

## 7. So, I Want to Tell My Coworkers I Have a Mental Health Issue

*Pick Your Time Before It Picks You*. Many times, crisis causes you to feel forced into a conversation. If you have an opportunity, then choose when and where you want to talk about this topic. Find some place quiet to have your conversation where you can feel composed and relaxed.

*Practice Makes Perfect*. Running your conversation through your head prior to actually having it can help you work out your kinks, but also, it might calm you because you've taken away some of the sting of the topic itself. Better still, if you have a friend who already knows your situation or you're working with a therapist, try practicing with them to make you feel more comfortable.

*Teachable Moment*. Likely, whomever you're revealing your diagnosis to doesn't know much about your mental health issue. Be prepared to answer some basic questions about what it is. In some cases, it might be helpful to have a few websites handy to fill in the holes.

*If Sorry is Needed, Say It*. One of the most difficult things for me to do was to go back and apologize for my behavior – not because it wasn't warranted. I had hurt some feelings and made some people feel uncomfortable, but once I found out my diagnosis, there was a part of me that felt like, "Hey, that wasn't my fault. It was the disease."

For me, I learned to adjust to the idea that while this disease wasn't me, I was still responsible for my actions that were a direct correlation from it. Throwing my hands up and saying, "hey, it's not my fault," was not an acceptable answer. While I did not chose this disease, I was not going to become a victim of it and let it lead me around.

And, with that, I needed to accept responsibility even if I couldn't always have control.

*Adjustments You May Need but Not Know About*. Reponses to your revelation will differ, but funnily enough, the ones I had the hardest time with were "What can I do to help?"

The first time I was asked that, I wasn't exactly sure how to respond. I knew some superficial things I had done to help self-regulate, but those were more instinctual. I didn't know for sure.

As I talked about it, I began to read more about adjustments at home and at work that would help regulate my cycling. If asked, I would recommend talking about experimenting with a few things to see how you respond. Just because it worked for someone else, doesn't mean it'll work for you. Alternatively, just because it didn't work for someone else doesn't mean it won't work for you.

## 8. What Are Typical Symptoms? (And How Can I Adjust for a Coworker?)

NOTE: Symptoms varying even within patients with the same diagnosis.

A few symptoms you may see in others that may indicate they made need adjustments:

### 8.1 Impulsiveness

*Issue:* Adults with ADHD, Asperger's Syndrome, autism, etc. may struggle with impulsivity and temper outbursts in the workplace. Many with these mental issues may interrupt, talk out of turn, abruptly end a conversation, walk out of a room unexpectedly, for instance. During a presentation by a Marine, my then-teenager with Asperger's Syndrome rose up, walked across the room in front of the speaker, and went out the door. When the teacher came to check on him, he told he that he simply didn't like that the speech had turn to a recruitment pitch and he didn't see the point in staying since he couldn't join the military anyway.

*Possible Adjustments:*

- **Create** a team building around role playing, especially if your team member needs to be prepare for a difficult conversation.
- **Embolden** team members to try relaxation and meditation techniques.
- **Encourage** your team member to write down triggers. Consider ways – together – to diffuse these situations.
- **Try** recognizing when you feel compelled to interrupt. Tap your knee with one finger three (3) times slowly to help regulate the compulsion.
- **Keep** a pad and pen handy to write down things you want to talk about, but would disrupt the flow of the current conversation. This will keep you focused on the current conversation without constantly reminding yourself what you want to talk about without listening to what is being said.

### 8.2 Hyperactivity

*Issue:* Adults with the hyperactive presentation of ADHD often do better in jobs that allow a great deal of movement, such as sales. For sedentary jobs, they may fidget, rock, tap, or do other activities that allow them to remain partially active. In most cases, they will not be aware of these activities, though others in the room may be very aware and annoyed by the actions. Many team members demonstrate this symptom in particular. Fidgeting, tapping, drawing may appear in multiple sit-down meetings. A bipolar disorder team member once described it as though she had two brains: one she would occupy with drawing while the other listened intently to the demonstration.

*Possible Adjustments:*

- **Remind** team members to take breaks, even simple ones, like standing up at their desks, or walking around the office for a few moments. This activity encourages oxygen back into the body and helps regulate hyperactivity.
- **Prompt** team members to bring ADD or fidget toys to meetings, or take notes in meetings to prevent restlessness.

### 8.3 Distractibility

*Issue:* Problems with external distractibility (noises and movement in the surrounding environment) and internal distractibility (daydreams) can be the biggest challenge for adults with ADHD. In fact, many workers with mental health issues complain of being overloaded with incoming information to the point that it can aggravate or generate an episode.

*Possible Adjustments:*

- Notably, **distractibility** can be one of the most difficult issues to deal with – both at the team member's desk and in meetings.
  - A private office, a quiet cubicle, or an unused conference rooms offers a chance away from the distractions that may acerbate the situation, causing frustration for everyone.

    - Additionally, consider allowing him/her to work from home.
    - A sporadic, planned calendar of when the team member can have uninterrupted time may help with anxiety and distractions that cause the work to go on longer or go unfinished.
- Another consideration, **create** a flexible schedule so your team member can have a few undistracted hours in the morning or afternoon.
- **Encourage** noise-cancelling headphones. Consider soft music or other white noise that may cover distracting noises from a nearby conference room, printer, call center, etc.
- As a team member, **consider** forwarding your calls directly to voicemail. Then, set aside a specific scheduled time to listen and respond.
- When writing, **turn off** the grammar and spell checker. Often when writing, these features can become disruptions, too. You can always turn them on after you've finished your first draft.
- **Carry** a small notebook to write down quick ideas so you can quickly return to the original task without contact switching.
- **Avoid** contact switching/multi-tasking.
- While in a meeting, **consider** ADHD-specific fidget gadgets that keep your hands busy, but your mind focused.

## 8.4 Poor Memory

Issue: Failing to remember deadlines and other responsibilities can antagonize coworkers, especially when working on a team. Because many of these workers struggle with processing every ounce of information that comes into view or earshot with little way to filter out unnecessary information, some information may get lost in the shuffle. Short-term memory is bombarded with so much metaphorical and literal noise; it struggles to determine what is worth retaining. Much like overloading yourself while carrying books, if you stumble, you likely won't have much finesse in keeping the most important books in your grasp.

*Possible Adjustments:*

- **Use** tape recording devices or take copious notes at meetings.
- **Write** checklists for complicated tasks.
- **Use** a bulletin board or computer reminder list for announcements and other memory triggers.
- **Rather** than use a cell phone (which can cause issues with impulsivity and distractions,) consider wearing a watch and using a printed or written day planner.
- **Write** notes on sticky pads and put them in a highly visible place.

## 8.5 Boredom

Issue: Because of their strong need for stimulation, some adults with ADHD become easily bored at work, especially with detailed paperwork and routine tasks. Many find it easy to create new ideas or designs, but have difficulty completing the seemingly mundane tasks to actually finish the work.

Possible Adjustments:

- **Give** yourself a time limit – either in duration or an alarm set to a specific time. This may help you stay on task.
- As we know in Agile, **breaking** up long task into shorter ones can help make the task seem more manageable, and help with feelings of progression.
- Take **breaks**, drink water, get up and walk around.

## 8.6 Time management

*Issue:* Managing time can be a big challenge for adults with ADHD.

*Possible Adjustments:*

- Like we do with epics and stories, **break** large projects into smaller pieces, with self-imposed due dates.
- Every time you reach a due date, **reward**
- **Consider** how much time you need to get to your meetings and then set your computer to beep beforehand. For

instance, someone people prefer a 15-minute warning before a meeting so they have a chance to change their mindset and then travel to the meeting with a fresh and open mind.

- **Think** about setting all the meetings you create to end early, leaving you time to refocus before going into another meeting. For instance, I set my hour-long meetings to end at 50 minutes after they start. For meetings of 30 minutes, I only schedule them for 20 minutes.

## 9. The Flying and The Thud

What is magical in my head, what keeps me awake nights under stars that disappear into the sun's light, would annoy and discombobulate others. But, as my friend who is color-blind says when people ask him does he miss not seeing all the colors in the spectrum, "This is the color I've always seen the world."

And that's me.

From the start, I've not had a day on this planet that wasn't washed in either the gorgeous Flying or the acidic Thud. The tectonic movements I feel as I walk, I've adjusted to their movements; others would stumble, maybe falling into the crevices created.

As I grow older and my body's chemicals naturally shift, so do the shape of the Flying and Thud. The dust devils that spiraled me from earthbound to an inner heaven revolve more leisurely. The decaying migration of depression crumbles like hardening lava, eking out acrid sulfur to remind me that I can avoid parts of my inner volcano, but it exists.

I will still hesitate to open an email with new edits on an essay. I will still avoid phone calls from friends who enjoy drama's vortex. I will still spin and spin and spin internally at the mere thought of joy.

I've learned ways to anticipate the shifts in the earth. I've learned to avoid things that open my inner crevices. I've learned to not look over the precipice.

I've learned – more importantly – what I have is not who I am.

What I do about it, is.

REFERENCES

[1] https://www.usatoday.com/story/tech/news/2017/04/27/is-uber-culture-to-blame-for-an-employees-suicide/100938330/
[2] SAMHSA, Center for Behavioral Health Statistics and Quality, National Survey on Drug Use and Health, http://www.samhsa.gov/data/population-data-nsduh/reports?tab=38

[3] https://profiles.nlm.nih.gov/ps/retrieve/ResourceMetadata/NNBBHS

[4] Kessler RC1, Adler L, Barkley R, Biederman J, Conners CK, Demler O, Faraone SV, Greenhill LL, Howes MJ, Secnik K, Spencer T, Ustun TB, Walters EE, Zaslavsky. "The prevalence and correlates of adult ADHD in the United States: results from the National Comorbidity Survey Replication." AM, Am J Psychiatry. 2006 Apr;163(4):716-23.

[5] de Graaf R1, Kessler RC, Fayyad J, ten Have M, Alonso J, Angermeyer M, Borges G, Demyttenaere K, Gasquet I, de Girolamo G, Haro JM, Jin R, Karam EG, Ormel J, Posada-Villa. "The prevalence and effects of adult attention-deficit/hyperactivity disorder (ADHD) on the performance of workers: results from the WHO World Mental Health Survey Initiative." J. Occup Environ Med. 2008 Dec;65(12):835-42. Epub 2008 May 27.

[6] Shoji M, Satoh S, Oda S, Kubota H, Imai Y. "Mental health in software engineers. II. Classification of occupational stressors, and relationship between occupational stressors and psychiatric disorders." Sangyo Igaku. 1990 Jul;32(4):258-64.

***

To read this article online with embedded hypertext links, go here:

**https://www.agilealliance.org/resources/experience-reports/the-flying-and-the-thud-mental-health-issues-on-agile-teams/**

# About Cassandra (Cass) D. Van Gelder

**Cass** started in theatre, learning theatrical and improv games. As an Agile Coach, she uses them to discover what’s holding teams back – individually and collectively.

Based in Vienna, Austria, Cass creates games, speaks internationally about mental health, and writes about subjects others avoid. Her latest projects include: Mom UniverCITY; “Presenting When You’d Rather Do Laundry” series; and her book "Creating Your Own Agile Games."

She trained in New York at Hunter College, taking Master Classes from Ruby Dee; worked at Second City in Las Vegas, studying with Jason Sudeikis ("SNL," "Horrible Bosses,") Kay Cannon ("Pitch Perfect" series, "New Girl,") and Joe Kelly ("SNL, "How I Met Your Mother."); appeared in large productions in New York, San Francisco, and Las Vegas, including the North American premiere of "Jerry Springer: The Opera" at the MGM Grand.

She, also, tends to be funny. Usually.

Write her at AgileIsMyName.O@yahoo.com or follow her Twitter account @AgileIsMyName_O.

# Don't Get Caught Holding The Bag

By Robert Weidner

Posted on January 29, 2018

The goal of Agile is really quite simple: to embrace change in pursuit of higher value. So how do we embrace change? That is also rather simple. To embrace change, we must lower the barriers to it. Therefore, we must make the cost of change low. How do you know if you're succeeding at Agile? It should be relatively easy to change course, and the cost of doing so should be far less than the cost of not doing so.

Some scaling frameworks fail to reduce the cost of change. If you're using one of these frameworks, you might be doing some Agile-like things, but you're not really Agile. You've simply bought into the marketing hype. It's okay, we've all been there. That only-sold-on-TV-for-a-limited-time towel that doesn't really soak up an entire swimming pool worth of water. Or that too-good-to-be-true investment that turns out to be a pyramid scheme. It happens to the best of us. But sooner or later, we figure it out.

The simple truth is this: some Agile frameworks are more Agile than others.

Smart CEO's are waking up to the fact that – when they bought SAFe – they were sold a bag of goods. The thing is, they were never buying SAFe… they were buying Agile. More than that… they were buying results. Agile merely served as the means, not the end. Then a wolf disguised as a salesman showed up and told them there was a cookie-cutter, fail-safe way to do this. They sold them on the idea of Agility. Only later did these CEOs realize they were left holding the bag… and what the bag was filled with.

Everyone seems to be looking for a pre-defined model they can implement that allows them to map current organizational structures directly into the new model. A few terminology changes, and viola… we're Agile! Or rather, we think we're Agile. But what have we really changed? Same hierarchy. Same program-based view of the world. Same quarterly planning cycles. Same functional silos. But wait, you exclaim... "Now we can see our dependencies!"

How many companies have tried to copy the Spotify model? It's a funny thing: the only company who no longer seems to use the Spotify model, is Spotify. They said it worked for them during that particular place and time, based on the size of their organization, and where they were in their journey. They've since outgrown the model, and the ways in which they work continue to evolve with them.

There's a reason the best companies in the world are the best, and I can tell you this: it's not because they use SAFe (they don't). So, what do they do?

After interviewing people from Google, Amazon, and elsewhere; I had the following key takeaways:

1. They measure business outcomes, rather than what Eric Ries has called "vanity metrics".
2. They do not permit "scaling".
3. They do not mandate Agile, nor do they have to. Many of these companies don't even use the word. It is a universally recognized best practice within these organizations. It is simply — as one person intoned to me — "the way software gets built".
4. The "business" and the "technology" are one and the same. They are product technology companies... not companies that have front-end business units supported by back-end IT.

The best companies figure out their own identity, their own path to agility. They don't copy someone else. They cut their own trail, and the journey never ends.

Agile isn't a destination, it's a way of being. Some people and companies have it. Most do not. You can't buy your way into it. But you can learn it. You just need a mentor who gets it. Not a vendor who converted their Program Management Office into an Agile Transformation Team when the market got hot, but someone who lives and breathes it. Not just the words, but what the words really mean. You need an authentic teacher, not a salesman. That is, if you truly want to become Agile, rather than an unemployed CEO holding a bag of poo. Granted, I've heard the severance packages can be decent.

***

To read this article online with embedded hypertext links, go here:

**https://www.livemindllc.com/post/don-t-get-caught-holding-the-bag**

# About Robert Weidner

**Robert** has over 20 years of experience in adaptive product development, leading large-scale organizational redesign to promote business agility. During this time, he's helped numerous companies — from startups to the Fortune 5 — implement customer-centric software development practices. From training to executive coaching, Robert has worked with hundreds of Product Teams to remove waste and optimize flow.

Learn more at www.LiveMindLLC.com.

# Anti-agile: Trick your partner into doing your laundry

By Adam Weisbart

Posted on December 18, 2018

Are you robbing your team of the chance to self-organize by being “helpful”?

Even with the best intentions, you may be.

Take my wife Erin for example… She’s fantastic. We met in San Francisco while she was finishing her PhD in molecular biology (read: she’s smarter than me), and she’s an amazing mom to our two-year-old.

“Hey Erin… I can’t remember… What temperature should I wash this shirt in? I don’t want to mess it up.”

Having answered this question before, Erin rolls her eyes a bit, and says “Just give it to me… I’ll take care of it for you.”

**Mission accomplished!**

With one simple question I no longer have to do my laundry.

Welcome to the wonderful world of learned helplessness!!

Am I an adult? Yes. (Allegedly)

Am I able to do my own laundry? Yes. (Well, once I made a load of whites pink due to a red sock. But YES!)

Do I have to do laundry? NO! I’ve mastered the “one simple question” trick! Ask it, and my laundry gets done for me. Not only do I NOT have to do my laundry, I don’t have to take responsibility for accidentally turning white loads pink. It’s a win-win (for me-and-me!).

At work, this learned helplessness often shows up as a “demotivated” person or team. I’d suggest that most of the time that person isn’t demotivated, they just know if they shrug their shoulders and wait, someone will jump in to fill the void.

If you’re like me, you want your team to be successful and happy so sometimes you jump in to “do their laundry” instead of letting them self-organize to get it done themselves. Maybe you're asking “who’d like to go next” during the daily scrum, or saying “I’ll go talk with the other team about us needing their help next week.”

Inquiring minds want to know: how do you help your team break this pattern of learned helplessness? Can teams really be expected to do their own laundry?

Certainly.

I suggest an experiment: After your next daily scrum, read this email to your team. Explain that you’d like to stop jumping in when you’re not actually needed, and ask them to help you be a better scrum master or coach by saying the following phrase when they feel like you're “helping”:

“Please don't do my laundry.”

I also recommend you ask them if you may utter the following phrase if it seems like they are abdicating their responsibility:

“I don't want to do your laundry.”

If these phrases seem too silly to you or your team, pick your own. Or better yet, keep them as-is and just blame them on me.

Whatever phrase you use, help each other stay accountable for creating an environment in which self-organization can occur.

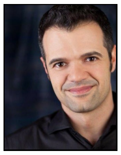
(By the way, I just had my wife proofread this post, and she informed me she doesn't want to wash my shirts for the certified scrum master course I'm teaching tomorrow. Thanks everyone. Thanks.)

***

To read this article online with embedded hypertext links, go here:

**https://www.linkedin.com/pulse/anti-agile-trick-your-partner-doing-laundry-adam-weisbart/**

# About Adam Weisbart



**Adam** Weisbart's humorous approach to the serious work of organizational change helps teams and individuals break out of old patterns and discover new ways to improve. His belief that hard work need not be a somber affair infuses everything he does.

Adam started his career as a software developer and went on to build a successful development firm in the early days of the web. Later, as an engineering manager in Silicon Valley, he came to terms with the fact that he was a relentless control freak. Thankfully, he discovered agility and liberated his teams. He has focused on mentoring high performing self-organizing teams ever since.

Today, Adam uses his personal experience of making the leap from traditional management to Scrum to help organizations make the same transition. He is the creator of Build Your Own Scrum (weisbart.com/byos), and the evil mastermind behind the retrospective-in-a-box subscription service Recess (RecessKit.com). His popular video "Sh!t Bad Scrum Masters Say" has been viewed hundreds of thousands of times — showing us all what NOT to do when adopting Scrum.

Adam is a Certified Scrum Trainer who teaches both Certified ScrumMaster and Product Owner courses around the world, regularly speaks at conferences, coaches teams, and helps people he hasn't gotten a chance to meet yet through his podcast Agile Answers.

If you'd like to discover how to use Adam's techniques to make your retrospectives hilariously awesome, you can grab a free copy of Adam's Agile Adlibs for use in your next retro:

https://weisbart.com/free

# Self-organisation in the sunshine: there's more to agile than post-its and stand ups...

By Steve Wells

Posted on May 24, 2018

giffgaff company meeting - a test of self-organisation?

Anybody else getting bored of posts about "agile stuff" like burn down charts, velocity variation, scrum certifications and other *minutae* of agile processes and methods? I wonder how agile companies who concentrate on these things, or simply care about them, really are, and if their digital transformations have changed much. Do they just use Scrum as a veneer, but still do "projects" that are successful if "on time and to budget" (instead of whether they delivered customer value or not) or just go round and round the scrum mouse wheel getting things to "done" but not delivering frequently, not constantly evaluating feedback and not measuring delivered value? What would we see if we went into one of the companies where this is the case and tried an experiment?

We recently (inadvertently?) tried an experiment in self-organisation. It wasn't a formal experiment, and we kind of only realised afterwards that it was an experiment, but I would be interested to see what happened if we ran this in other companies. Is this, in fact, an experiment that measures how far the agile mindset is embedded in everyone in the company?

Let me explain...

Every month, we have a monthly meeting. This month's meeting "agenda" was "bring a water bottle and training shoes to the park, and we'll have a bit of a chill and some games in the sunshine". And that was it; nothing else.

So, we all trooped off to the park - yes, *everybody* - where there was rounders kit, mini football goals and a cricket set. Within a few minutes, everybody (100+ people) had self-organised into 3 games with sensible numbers on each side (well, the rounders teams were a bit large...) and started playing games having agreed a set of rules and even appointed a referee for the rounders and an umpire for the cricket. (We should have had a referee for the football - I was *never* offside :-) ). Self-organisation in action - teams are used to self-organising so that's just what they did; no sign-up sheets, no appointed team captains, no picking sides - they just got on with it. And literally everybody got on with it; the CEO was backstop at rounders, the CFO lining up beside the grads in one of the football teams. No hierarchy on the Rockingham Rec.! And all of this happened within a very short space of time.

However, it goes deeper than that. The history of this event is that these monthly meetings are organised by a different teams in the company each month. Usually, this means it will be based around company strategy, or some initiative, or some big idea or project status. Or something along those lines. This particular month, however, as it was Mental Health and Wellbeing week (and the sun was out), the organising team - and the Ministry of Sports (note 1) - decided to just get everybody out in the sunshine having a bit of a chill. The leadership team had suggested we might do "something like this" and suggested a couple of options, but that was it - from suggestion to implementation was a matter of a couple of days - no long winded planning sessions or sub-committees to work out the details, just take the idea and run with it with no further interference, and no further permission required.

This is where we realise how ingrained these agile/Teal values - empowerment, autonomy, trust, self-organisation and devolved decision making etc. - are within the company. This is why agile works here; the entire company has internalised these values, and this is our culture. It is the giffgaff way, plain and simple - we are curious and positive and we'll just go for stuff and see what happens.

I'd be fascinated to see how this would work in other companies. I asked someone, who will remain nameless, what would happen in their company, which must also remain anonymous (for obvious reasons), if we tried this. She surmised that "everybody would complain that they didn't know what was happening, and nobody was in charge. When somebody stepped up to take charge, they'd complain about who it was and how they were doing it wrong". I suspect some similar scenario might happen in many organisations, and it would be fascinating to try this as an experiment in a variety of companies with different cultures to see what actually happened.

If I can make that happen, I'll be sure to blog it, and do send me any instances where you've tried it...

Notes:

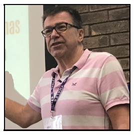
1. Along with our members, we are the size of small to medium sized country so a) we call ourselves the Joyous Republic and b) we need Ministries to run things; Sports, Good Deeds, Wellbeing, etc.

***

To read this article online with embedded hypertext links, go here:

**https://www.linkedin.com/pulse/self-organisation-sunshine-theres-more-agile-than-post-its-wells/**

# About Steve Wells



**Steve** Wells has been in technology for some time - he wrote his first program on punched cards in the 80s - and has worked in myriad coding, management and coaching roles for many years. Having been an agile, XP developer for many years, he struck out as Scrum Master and Agile Coach around 10 years ago and has since been heavily involved in transformation at Sky, M&S and, latterly, giffgaff. He is a keen proponent of OKR-driven Lean UX which, let's face it, is basically agile as it was originally envisaged before all these frameworks and certs took over, and speaks regularly on the subject. A regular blogger and speaker at conferences such as Agile Cambridge, Agile Business and CIPD, he is always ready to provoke discussion even if unpopular. Particularly with QAs and Jira lovers...

# Agile Exposes the Bad Boss

By Ed Wisniowski

Posted on October 22, 2018

I was getting on an elevator at the office and I decided to make small talk with someone as we were heading up to our respective floors.

A bad boss is just toxic.

“Ready to set the global economy on fire,“ I joked.

My fellow traveler got a gleam in their eye and said, “The flames are so colorful.”

I got off on my floor and breathed a sigh of relief. The metaphorical pyromaniac was too eager to be pulling my leg. The experience brought into stark contrast how tired many of us have become in the business world. The daily frustrations of working in a modern office force many professionals into the cynical behavior of inflicting harm on others as a means of satisfaction. It is perverse, and it is wrong. The cynicism in the elevator is one of the reasons I have been such an enthusiastic proponent of agile. I firmly believe there must be a better way to structure work so that it is sustainable, sane, and satisfying.

Inc. Magazine and Monster.com pointed out this week that 76% of bosses in business are “toxic.” This toxic leadership is why so many people rely on jaded cynicism. It is crucial as an agile coach and scrum master to break this cycle of toxicity. According to the article in Inc. magazine, a toxic boss exhibits some or all of the following traits.

1. They are power-hungry
2. They micromanager
3. They are absent
4. They are incompetent

It is up to people like me to expose these bosses to the organization and coach them to be better.

## The Power Hungry

Working for a power-hungry boss is a little like being a supporting cast member in Game of Thrones; you are going to wind up suffering a cruel ending to satisfy someone else’s ambition. It surprises me how many business leaders think servant leadership is similar to the game “Masters and servants.” The reality of servant leadership is much different. In the end, what everyone needs to understand is a power-hungry boss is concerned about one thing; themselves. A power-hungry boss will put personal interest over the needs of the company and employees. Agile exposes the power-hungry because they often become impediments to shipping solutions.

## The Micromanager

The hardest part of leadership is the lack of control we have over our fellow humans. A leader can spend years training people to do the right thing and meet a certain performance level, and they can still disappoint at critical junctures. To combat this helplessness, managers create processes and steps which they expect people to obey like robots. It creates an illusion of control where employees do what they can to avoid hassle rather than what is necessary to succeed. Thus, reports have perfect typography and proper tab spacing, but the data within that report shows lead conversion is falling. The emphasis on working solutions instead of comprehensive documentation in agile should expose micromanagers.

## The Absent

Over the years, we tell countless stories about military leaders who "lead from the front," instead of from behind a desk. I am currently reading one about William Slim who commanded the 14th Army of Burma during the Second World War. It is easy to get caught up in the trappings of authority. In an office of cubicles, having your office is a status symbol. It gives you the power to shut people out and focus on administrative duties. The autonomy and control over who has access is a powerful motivation for people to advance into leadership. In reality, a leader has to be more visible to the people they are leading. A leader should know about the people who make them successful. If the leader is not around and they become distant figure the people who make them successful will ignore them in time of crisis. Agile attempts to counter this kind of toxicity with its emphasis on face to face communication.

## The Incompetent Leader

A leader should not be able to do your job, but at the very least they should understand what it takes to do your job. What I have discovered over the years is people who have never managed a computer network or written a line of code often lead technology teams. These people know how to manipulate budgets and control the project, but they do not know how to direct technology professionals because they think they are no different than shipping clerks or factory workers. Agile with its emphasis on cross-functional teams and delivery exposes the incompetent.

I am a big believer in the idea that you should tell and expose the truth wherever you find it. Sooner or later, someone in a position of authority is going to act on that truth. I feel this way because it is how we defeated leaded gasoline and paint. It is how we have reduced smoking in the United States by half since 1964. It is an approach which led to the birth of agile.

If we are honest with ourselves, we should acknowledge the power-hungry, micromanagers, the absent, and incompetent and expose them so their toxic effect on the workplace can be mitigated. It matters, and if we are not successful, all we can do is watch the pretty colors as the world burns.

Until next time.

***

To read this article online with embedded hypertext links, go here:

**http://goeeethree.blogspot.com/2018/10/agile-exposes-bad-boss.html**

# About Ed Wisniowski

**Ed** Wisniowski is a technology veteran with over twenty years of experience. He has been working in the realm of agile for ten years and is a Certified Scrum Professional - Scrum Master and Certified Scrum Professional - Product Owner. He is currently the co-founder of the Agile Coaching Exchange in Chicago and a proponent of “*Healthy Ownership*,” for business teams. He has spoken at the Agile 2018 conference talking about “*Delivering Value and Defeating the Cobra Effect*.” He specializes in helping product owners become more successful and helping the business measure change.

He is currently working on becoming a Certified Team Coach with the Scrum Alliance.

In his spare time, Ed collects toy soldiers, plays board games, drinks craft cocktails, and likes all forms of cinema. He lives in the greater Chicagoland area with a house full of plants.

You can reach Ed via his blog goeeethree.blogspot.com or email ed.wisniowski1968@gmail.com

# Understanding the Kanban Guide for Scrum Teams

By Yuval Yeret

Posted on March 14, 2018

It’s been so exciting to hear so much positive feedback and interest in the new Scrum.org Kanban Guide for Scrum Teams and the accompanying Professional Scrum with Kanban class. Creating the class and guide together with Daniel (Vacanti) & Steve (Porter) and then working on getting it to market in a professional way (how else? ) with the Scrum.org staff has been a great experience and a major focus area for me in the last couple of months.

As you might imagine, together with the interest come some questions about some choices we made in the design of the guide and the class. Several are emerging as the frequently asked ones. I wanted to tackle a couple of those in this post.

## Where are some of the core Kanban practices?

The major one we’re getting comes from people who are experienced Kanban practitioners who notice that how we describe Kanban in the Kanban Guide for Scrum Teams is different than the definition they’re familiar with. (Including my Kanban Primer for Scrum Teams blog post for example…) This isn’t an oversight. It’s by design. When we set out to create the Kanban Guide for Scrum Teams/approach we had a specific context in mind. That context is teams using Scrum according to the Scrum guide, ideally professionally.

In the Kanban Guide for Scrum Teams, we focus on helping in this context. These teams already have a collaborative inspect and adapt experimentation process together with a set of explicit feedback loops they’re using. So, we set out to define the minimal simplest set of Kanban practices that these Scrum teams would need to add in order to achieve steadier, healthier, more sustainable flow (I like to say it is like moving from a sprint that looks like a swamp to one that looks like a river).

After some discussion we decided that these practices actually complement what a professional Scrum team is already doing :

- Visualization of the workflow
- Limiting WIP
- Active management of work items in progress
- Inspecting and adapting their definition of "Workflow"

While we agree with the importance of “Improve Collaboratively (Using models and the scientific method” and “Implement feedback loops” we think they are redundant in a professional Scrum context.

## Where are some advanced Kanban concepts like Classes of Service, Cost of Delay, Flow Efficiency?

They’re not part of the guide because we don’t consider them part of the “Minimally viable set of practices” a Scrum team should focus on when trying to improve their flow. Having said that, our guide and especially the PSK class provides people with some pointers towards advanced complementary Kanban/flow practices/metrics that at least some can use to continue their learning and improvement journey.

Beyond that - Some of them might be useful in some Scrum contexts, some less so.

## Is this an application of the Kanban Method or not?

In my personal view, it is pretty close, as long as you assume professional scrum is your starting point. (see a blog post I wrote back in 2012 about this context). You are starting with the way the team uses Scrum and with respecting their current Scrum process & roles. You are obviously interested in pursuing an incremental evolutionary change to improve your performance and satisfaction with your process beyond what you're currently achieving with Scrum. There is that argument that limiting your work in process is far from being an evolutionary change but rather a disruptive revolution. My personal take on it is that yes, limiting your work in process and moving to a disciplined pull mode is far from being easy, but it's still evolutionary compared to changing team structures, roles, process flows. And in any case, this is an argument about the Kanban Method outside of a Scrum context as well. A professional Scrum team should actually have an easier time limiting WIP than most.

## Is this ScrumBan?

Depends who you ask. Some people's definition of ScrumBan is "A way to help teams transition from Scrum to Kanban". This isn't what we're talking about here.

Another definition (that I subscribe to) is to see ScrumBan as a way to introduce Lean/Kanban flow into a Scrum context – while keeping the core Scrum process intact. This is pretty similar to our take on the process Scrum teams will typically take to get to an effective combination of Scrum and Kanban.

Finally, a variant on this definition is to see ScrumBan as simply the Scrum+Kanban combination itself, without worrying about your starting point and journey. This, in my view, is indeed what the Kanban Guide for Scrum Teams describes.

## Why/When should you add Kanban to Scrum?

The last question I want to tackle is one of the first you might want to think about. Essentially the question is "Why bother? Isn't Scrum great as is?"

Most of the teams I've worked with since 2010 or so find Scrum+Kanban to be the ideal mix. I've helped Scrum teams achieve an even healthier smoother flow by adding Kanban to their process. I've helped Kanban teams accelerate their pace of improvement by adding cadence/rhythm and clarity. I've helped teams look beyond their Sprint at the end to end flow all the way from idea to outcomes using a Kanban system. I've helped organizations manage the flow across several Scrum teams using a Kanban system.

When a Scrum team wants my opinion on whether adding Kanban would be a good idea I typically ask them to think about how hard it is for them to Sprint and whether they feel like they have good flow during the Sprint. (And like I mentioned above - do they feel like their process is a swamp or a river). It's as simple as that. I find most Scrum teams struggle to achieve good sustainable healthy flow and Kanban tends to help them with that.

## When is Kanban with Scrum a bad idea?

Some Professional Scrum Trainers asked "When would it be a bad idea to introduce Kanban to your Scrum? What are some indicators that you should stop using Kanban as part of your Scrum?" I can't think of any team where I thought they should stop using Kanban. If they understand Kanban and do it well, there's really little that can go wrong. The problems start when they don't understand Kanban or use it as an escape from the challenges of Scrum. Yes, Kanban can help you make your Scrum more sustainable and healthy, but please don't add Kanban if you're looking for an escape from the difficulties. Kanban done well adds discipline to your Scrum. Another bad time to introduce Kanban is when the team isn't looking to improve. If things are working well or more importantly if the team perceives things as working well, they won't have the energy needed to successfully add Kanban to their process. So make sure you agree on pains/motivations before you move forward to implementing something like Kanban.

## Kanban - A way back to Scrum!

To finish with scenarios where introducing Kanban IS a great idea – It pains me every time I see a team/company using Scrum as a new variant of “project management command and control” focusing more on tasks, story points, velocity and burndown than on empiricism leveraging Done Increments of potentially releasable product.

What I’ve noticed is that introducing Kanban ideas helps these teams/companies finally understand what Scrum really is about and shed a lot of the unnecessary and even harmful baggage and instead refocus on the transparency, inspection, and adaptation brought to life by the core Scrum events, roles, and artifacts. Pretty amazing isn’t it?

Interested to learn more about how Kanban and Scrum make each other better together? Join a public Professional Scrum with Kanban class or request a private training for your team.

***

To read this article online with embedded hypertext links, go here:

**https://www.scrum.org/resources/blog/understanding-kanban-guide-scrum-teams**

# About Yuval Yeret

**Yuval** Yeret is a Professional Scrum Trainer and SAFe SPCT at AgileSparks. Known as “Mr Kanban Israel” and a recipient of the Lean/Kanban community leadership award, He's a steward of the Professional Scrum with Kanban class and Kanban for Scrum Teams guide at Scrum.org. He’s been helping teams and organizations worldwide bring Scrum and Kanban together since 2007. He’s the author of “*Holy Land Kanban*” and “*SAFe from the Trenches*” as well as numerous articles on the AgileSparks Blog. You can reach him at Yuval@AgileSparks.com or on Linkedin.

# Optimizing internal product communications

By Daniel Zacarias

Posted on January 26, 2018

A lot of our time as PMs is spent communicating with other people in our organization. There's high-level communication and discussion, and there's low-level interaction with stakeholders about the day-to-day of the product's development and operation. Since the majority of our communication time is typically spent on the tactical, let's talk about that.

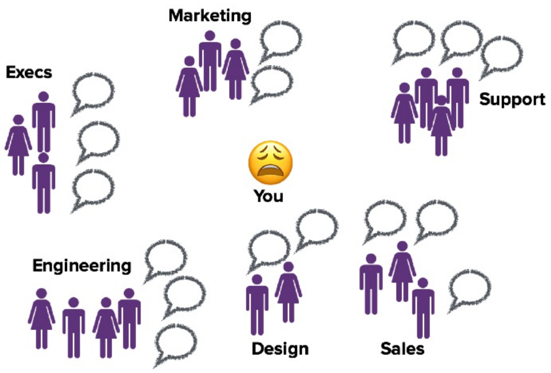
It seems that we're endlessly providing (or seeking) answers about anything related to the product. This happens naturally, because our role is at the center of the process that figures out what to build, builds it, and then gets it to market. The cross-functional nature of what we do means that **we have to align and inform everyone around us**.

*Most days, we're answering questions like:*

- *"How's that feature coming along?"*
- *"What's going to be on our next release?"*
- *"Why are we working on this?"*
- *"Why is that taking so long?"*
- *"Is that thing fixed already?"*
- *"How long will we be down for?"*
- *"Did you know there's a bug when people try to do 'so and so'?"*
- *"How's that KPI doing?"*

Or perhaps we're reaching out to other teams to ask questions like:

- *"Which bugs have been reported this week?"*
- *"What are customers asking for in our support channels?"*
- *"What are the top objections we're getting on sales calls?"*
- *"Which are the biggest problems in the product right now?"*



Granted, this isn't exactly the stuff of dreams nor the reason we went into Product, but it's still critical to our success. Efficiency-minded as PMs are, most of us have tried at least one of these tactics to handle the need of getting everyone on the same page:

- Simply answering as we go along, in order to save time on meetings and status updates. But then we get asked the same thing multiple times, people don't remember, or those that don't ask (but should be informed) are kept

out of the loop.
- Create internal dashboards, docs or wiki pages, only to realize that's where information goes to die, because people can't find it, don't understand the tool or forget to go there.
- Send internal email blasts with updates, but people complain that there's too much info in them, that they're too frequent (or not enough), that they "haven't read it yet", or something along those lines.
- Send updates in Slack, but then people say that there's too much going on there and they missed it, aren't in the right channel or weren't paying attention at the time.
- Set up sync meetings to get everyone aligned, but they take time to prepare and facilitate, they're not always useful or relevant for everybody and it's difficult to get everyone that needs the info to attend (either because they're out of the office or have "other things on their plate").

The end result is that we settle on something, spend a lot of time on it, knowing (or suspecting) that it's not really effective. These are problems that we typically accept as inevitable, but shouldn't.

If we stop to think a little bit about our internal communication strategy we can be both more effective in getting the message through, and reduce the amount of time we spend on this (having specialized communication tools for PMs would help even further, but I'm working on that.

## The communication dimensions

There's a simple exercise that brings a lot of clarity to how we can structure our communication about the product with stakeholders. It's not rocket science, but I've found that most of us get so caught up in the action that we don't really do it. It's about breaking down each bit of information we need to transmit (or get) into 4 components: **content**, **audience**, **timing** and **format**. In other words:

- **What** are we saying (or asking)?
- To **whom** should we say it?
- **When** should we say it?
- **How** are we saying it?

Let's go through each.

### Content (What)

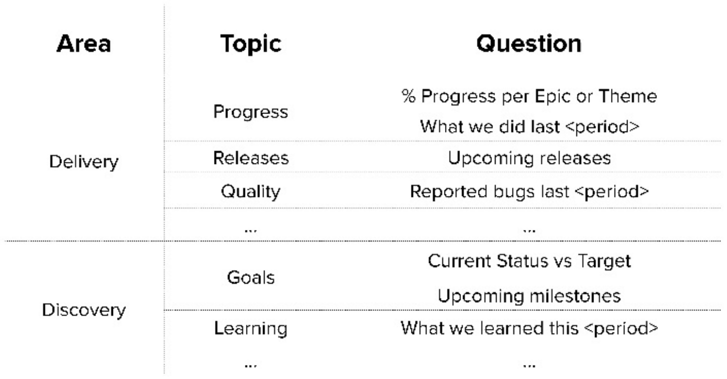
To start the exercise off, we need to write down the list of questions that we get asked by (or that we ask) stakeholders. The top of the article has a few of them, and you probably have some more like those. That's the easy part.

Where I find there's also a lot of value is in listing the questions that aren't being asked, but should be. That is, information that if other people (or yourself) had on a regular basis, many questions, conversations and discussions at both a high- and low-level, would be avoided/simpler/faster.

Most of the example questions above are about the **delivery side of product development**, and they come naturally. However, it's less common that teams have processes to periodically stay updated and aligned on what's happening on the **discovery side**. You know, questions like:

- *"What are we learning at the moment?"*
- *"What will be working on next (and why)?"*
- *"How are we doing with regards to our goals?"*
- *"What important events/milestones are coming up?"*
- *"What are customers telling us?"*
- *"What's working well and what isn't?"*

Including this type of questions into our "catalog" of internal communications definitely makes aligning different teams across the organization easier.

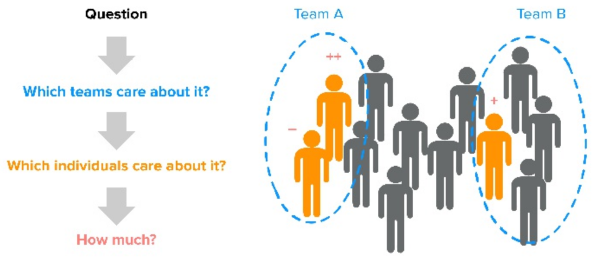


| Area | Topic | Question |
|---|---|---|
| Delivery | Progress | % Progress per Epic or Theme<br>What we did last <period> |
| | Releases | Upcoming releases |
| | Quality | Reported bugs last <period> |
| | ... | ... |
| Discovery | Goals | Current Status vs Target<br>Upcoming milestones |
| | Learning | What we learned this <period> |
| | ... | ... |

## Audience (Who)

Every question should have an audience, based on who’s asked you (or should know) about it. This also includes yourself: you can be the audience of topics that you’d like to know about from other teams (usually customer-facing ones), which can bring you valuable feedback about the work your team is shipping (or considering). Think about it as either inbound or outbound comms.

To make this easier, I prefer to start by identifying the team or group of people that should know about each topic. Then, to prevent the risk of sending info to people that don’t need it, I narrow it down to the key people in those groups (usually the ones that are most involved with the product). Finally, classify them over the level of interest in the subject (e.g. high, medium, low, or ++, +, -).



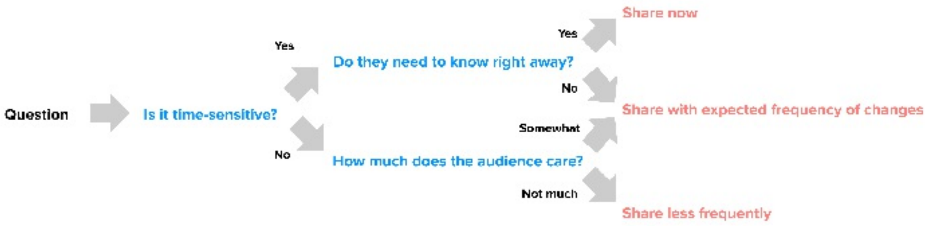
## Timing (When)

There are two kinds of information when it comes to timing:

- What needs to be shared as soon as something happens
- What looks to a past or upcoming period of time to answer a question

Few things actually need to be shared right away with our stakeholders. The most obvious ones being new releases or some kind of critical bug or downtime. This means that the “When” variable for some of our questions might be “right away”, but we should always consider how often the event occurs and how much the audience cares about it. If the answer to those is “happens frequently” or “doesn’t care that much”, then it’s preferable to group these events and report on them over a period of time.

Most of what we need to inform or ask about is not immediate and is related to “what’s coming ahead” or “what’s happened” over a certain period of time. Which is good, since there’s a certain cadence to product development. Things happen within (somewhat) predictable frequencies: development cycles, planning, releases, strategy/goal setting, experiment results, etc. Stakeholders that are more invested in a particular topic should be kept updated at the frequency at which it changes; others can be updated less frequently.

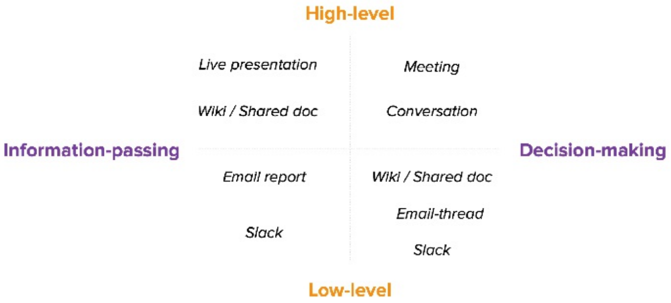

Question
Is it time-sensitive?
Yes
No
Do they need to know right away?
How much does the audience care?
Yes
No
Somewhat
Not much
Share now
Share with expected frequency of changes
Share less frequently

Whatever timing you choose for each bit of information you transmit, be consistent. It's important that you keep to a schedule, even when there aren't changes in what you're sharing. People need to trust your "system" and that they'll reliably get the info they care about. When they don't, they'll resort back to asking you directly about it.

### Format (How)

This is the trickiest variable to determine, because it depends a lot on the organization's culture and ultimately, each individual stakeholder.

However, there's a way to look at this that I've found useful. Think about your communication's *intent* and *granularity* as two orthogonal dimensions. The Intent of the communication can range from information-passing to decision-making. *Granularity* ranges from low- to high-level–that is, are you talking about execution or about top-level goals and plan? If you plot this, then which formats to use becomes more evident.



Although this post focuses on the information-passing side of things, it's useful to consider the difference in formats as we go into a different type of communication like decision-making.

Let's take meetings as an example. We're on them all the time. There are literally thousands of books on the subject: why they suck, why they're great, how to lead them successfully, etc. Regardless of how we feel about them, they're the most accessible communication tool we have for discussions on topics that aren't fully structured yet (e.g. decision-making)–that's where they add the most value. On the other hand, if all we're doing is regularly getting together to go around the table and share updates with everyone else, they don't add much value, especially if we can automate the process (getting data out of some app or using a tool like Basecamp).

## Putting it together

The exercise led us through these steps:

1. List all the topics we're asked, we ask others, or that we want people to know about
2. List everyone who's interested in each topic
3. Find the right timing for each topic and audience based on how often it changes and the audience's level of interest
4. Figure out the most effective format for the content and audience to which we're communicating

This is in itself valuable, because it leaves us with a map that shows:

- The scope of internal communication work that (ideally) needs to happen;
- How much we are communicating with each person or team.

However, the real value is that with this information in hand, we can:

- Prioritize what to say to each audience, so they're not overloaded and get updated about what's most relevant to them;
- Prioritize the use of our time based on the effort that each topic requires and the value it provides to its audience–some topics probably "aren't worth it";
- Control different variables to test what's most effective in getting the message across;
- Be aware of the topics that are being communicated and those that aren't, so that we're prepared to handle those questions as they come up.

## What's next?

Over the next few posts, **I'll be sharing some tips and templates that can help you implement this in practice**.

Also, if you're a newsletter subscriber, you'll get **early access (and a special discount) to a tool that I'm launching soon**, designed to **save Product Managers time in their updates to the rest of the company**.

***

To read this article online with embedded hypertext links, go here:

**https://foldingburritos.com/articles/2018/01/26/optimizing-internal-product-communications/**

# Report templates for fewer status meetings

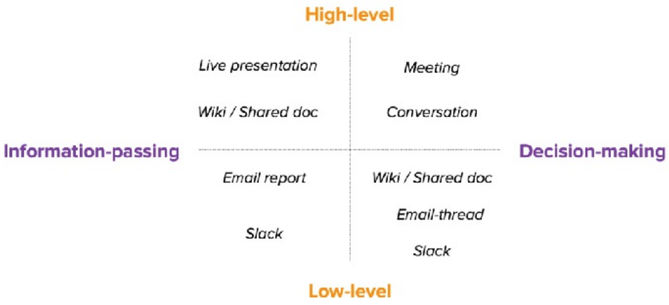
By Daniel Zacarias

Posted on February 13, 2018

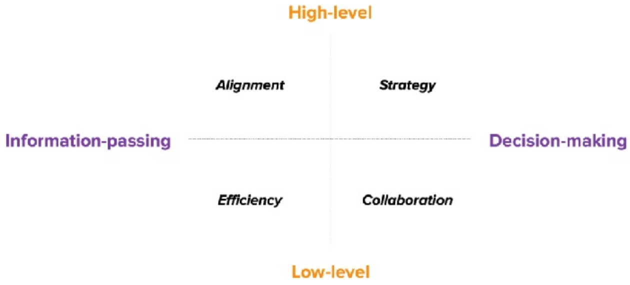
In my previous post, I described a way to think about our internal product communications, and how that can help us optimize our time and make us more effective as Product Managers. The main goal was to show that **by thinking about what**, **when** and to **whom** we need to **communicate**, **we can find different ways to get the message across without spending so much time on it**.

This post is about making these ideas concrete through a set of guidelines, templates, and JIRA+Excel tips so you can create effective status and progress reports quickly, **have fewer meetings**, and **get out of the building**, which is where you need to be.

Out of the key dimensions (**Content** / *What*, **Audience** / *Who*, **Timing** / *When* and **Format** / *How*), the Format is probably the hardest to get right. You may recall this graph showing which formats are more helpful for each type of communication we need to have:



One of the reasons that we as PMs feel like we're always "deep in the weeds" is that we're spending way too much time sharing status and progress updates with other teams and stakeholders, most likely in meetings. You'll notice however that there isn't much space for meetings in this graph. The reason is simple, when you think about what each type of communication is for:



- High-level decision-making is about discussing the big picture, setting our product's **strategy**
- Low-level decision-making is about the day-to-day execution, **collaborating** cross-functionally to build the product
- High-level information-passing is about informing others about our goals' status and progress, creating **alignment**
- Low-level information-passing is about keeping other updated about our work in progress and releases, so they're able to do their jobs with more **efficiency**

The problem is that **the sheer amount of things that happen on the left of the graph (specially on the lower side), easily drives us into being part-time project managers, trying to keep everyone informed about what's going on**. There's always a lot happening in the recent past and near future of the product. These are some of the biggest areas that stakeholders care about within this space…

**Work / Progress**

- *What are you working on (and why)?*
- *What progress are you making on X?*
- *Anything we need to be concerned about?*
- *What have you worked on recently?*

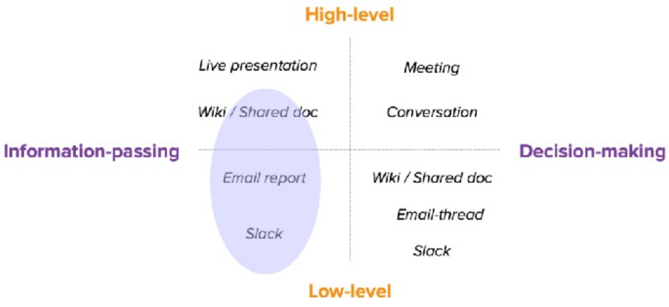
**Releases**

- *When are our next releases happening?*
- *What progress are you making on them?*
- *Anything we need to be concerned about?*
- *What have you released recently?*

**Goals / KPIs**

- *What's the current status of our goal/KPI?*
- *How is it moving (vs target)?*
- *Why is it moving like that?*
- *Anything we should know or think about?*

If we plot these on our graph, they'd occupy this space:



All of the questions above are **about keeping people updated**, but **they're not about discussion**. Our goals and work are defined elsewhere, at different times, and there are good tools for that (like face-to face discussions, presentations, and roadmapping, product design and collaboration tools). However, to answer these day-to-day, follow-up type-of-questions, **most of us end up scheduling regular status meetings, because "they're just easier"**. I believe **this is a waste of everyone's precious time**, and **we should save our meeting time for when it's most effective:** when we need to decide things, not when we're just sharing updates.

# Templates and tips for fewer status meetings

One of the main reasons we set up a meeting instead of sharing a report with stakeholders is that our project management tools require some work from us to get the data into a format that's actually useful to them. Another reason is that people from outside of Tech don't really like / know / want / remember to go to JIRA and Confluence.

Hence, it's critical that you choose a communication medium where you know they'll have easy access to the information. In most places, that's still email. In others, it might be Slack, or a shared document/wiki page with a simple format that people know how to get to (and that you regularly remind me them to have a look). We have to **regularly reach stakeholders where they are**, and make it **extremely easy for them** (not us) to consume the information.

Let’s now go over some report templates that I’ve found effective in addressing the questions above, along with some JIRA+Excel tips, that should help you get them ready more quickly. This post focuses on the Progress and Releases side of things because they share the same source and their reports can be produced in similar ways. Goals/KPIs will have to wait for a future post.

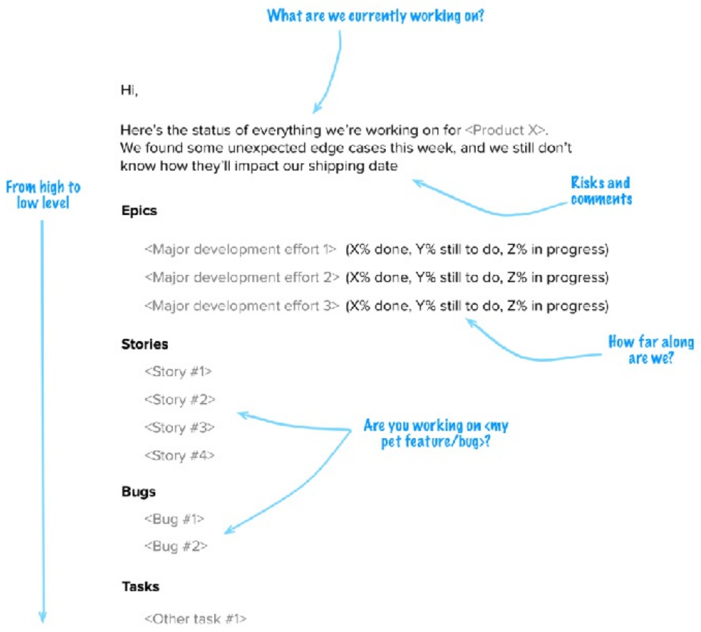
Click here to download the spreadsheet and follow along the examples (by the way: it’s built with a more limited Excel feature set because I’m on a Mac – if you’re on Windows, you could play with other PivotTable features like Timelines).

## Work

There are two common views that stakeholders need into our work: the progress or status of what we’re currently doing, and what we’ve recently completed. Remember the **timeframe we’re addressing** here: **present**, **recent past** and **near future**. Discussions about the future–that is, what should get higher priority in the roadmap and backlog need to be separate, and a good use of meetings and presentations–as they’re related to the product’s strategy.

### Progress / Status

To some, this is represented by the progress we’re making on an *Epic*, *Theme* or *Initiative*. Other stakeholders also need to know about more low-level tasks that are in progress. This report template addresses these views and common questions around them:



Hi,

Here’s the status of everything we’re working on for <Product X>.
We found some unexpected edge cases this week, and we still don’t know how they’ll impact our shipping date

**Epics**

<Major development effort 1> (X% done, Y% still to do, Z% in progress)
<Major development effort 2> (X% done, Y% still to do, Z% in progress)
<Major development effort 3> (X% done, Y% still to do, Z% in progress)

**Stories**

<Story #1>
<Story #2>
<Story #3>
<Story #4>

**Bugs**

<Bug #1>
<Bug #2>

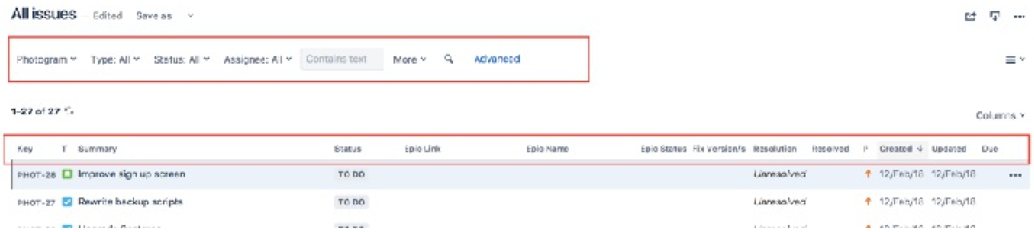
**Tasks**

<Other task #1>

What are we currently working on?
From high to low level
Risks and comments
How far along are we?
Are you working on <my pet feature/bug>?

#### How to create this with JIRA and Excel

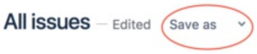
In your JIRA project, select “All Issues” and make sure that all issues from any status are displayed, and that the first few columns are in this order (the Excel spreadsheet assumes they’ll be in those positions, but you can later customize the filter and spreadsheet):



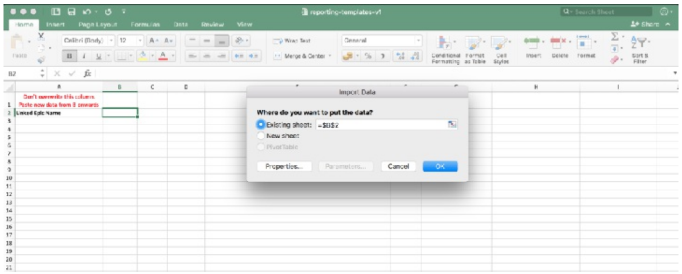
1-27 of 27

| Key | T | Summary | Status | Epic Link | Epic Name | Epic Status | Fix Version/s | Resolution | Resolved | P | Created | Updated | Due |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PHOT-28 | □ | Improve sign up screen | TO DO | | | | | Unresolved | | | 12/Feb/18 | 12/Feb/18 | |
| PHOT-27 | ☑ | Rewrite backup scripts | TO DO | | | | | Unresolved | | | 12/Feb/18 | 12/Feb/18 | |

The reason to include all issues in any status is that you can still easily filter them out in Excel, whilst making it easier to calculate an Epic's progress based on its composing issues' statuses. Save this filter so you don't have to do this again later and export the results in CSV format.



Clear the contents of the Issues tab in this spreadsheet (starting from column B, because A is reserved for a formula that helps us out later), and import the CSV.



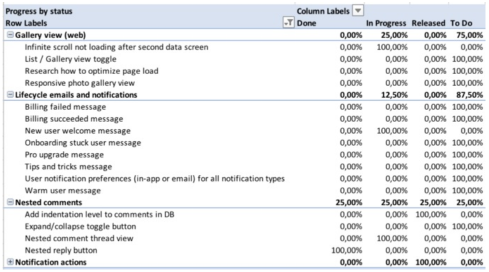
Don't overwrite this column
Paste new data from B onwards
Linked Epic Name

Import Data

Where do you want to put the data?

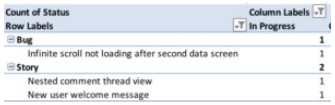
● Existing sheet: =$B$2
○ New sheet
○ PivotTable

Now, go into the "Progress – Epics" tab and refresh the pivot table. You'll have a list of all epics, showing their % progress by status: i.e., the % "To Do", "In Progress", "Done" or any custom statuses you may have.



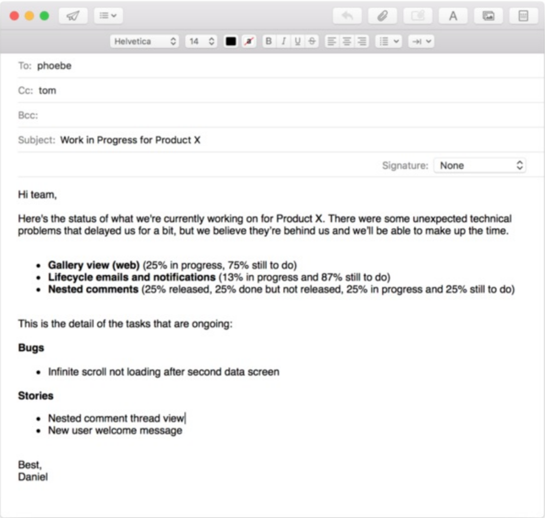
| Progress by status | Column Labels | | | |
|---|---|---|---|---|
| Row Labels | Done | In Progress | Released | To Do |
| ⊟ **Gallery view (web)** | **0,00%** | **25,00%** | **0,00%** | **75,00%** |
| Infinite scroll not loading after second data screen | 0,00% | 100,00% | 0,00% | 0,00% |
| List / Gallery view toggle | 0,00% | 0,00% | 0,00% | 100,00% |
| Research how to optimize page load | 0,00% | 0,00% | 0,00% | 100,00% |
| Responsive photo gallery view | 0,00% | 0,00% | 0,00% | 100,00% |
| ⊟ **Lifecycle emails and notifications** | **0,00%** | **12,50%** | **0,00%** | **87,50%** |
| Billing failed message | 0,00% | 0,00% | 0,00% | 100,00% |
| Billing succeeded message | 0,00% | 0,00% | 0,00% | 100,00% |
| New user welcome message | 0,00% | 100,00% | 0,00% | 0,00% |
| Onboarding stuck user message | 0,00% | 0,00% | 0,00% | 100,00% |
| Pro upgrade message | 0,00% | 0,00% | 0,00% | 100,00% |
| Tips and tricks message | 0,00% | 0,00% | 0,00% | 100,00% |
| User notification preferences (in-app or email) for all notification types | 0,00% | 0,00% | 0,00% | 100,00% |
| Warm user message | 0,00% | 0,00% | 0,00% | 100,00% |
| ⊟ **Nested comments** | **25,00%** | **25,00%** | **25,00%** | **25,00%** |
| Add indentation level to comments in DB | 0,00% | 0,00% | 100,00% | 0,00% |
| Expand/collapse toggle button | 0,00% | 0,00% | 0,00% | 100,00% |
| Nested comment thread view | 0,00% | 100,00% | 0,00% | 0,00% |
| Nested reply button | 100,00% | 0,00% | 0,00% | 0,00% |
| ⊞ **Notification actions** | **0,00%** | **0,00%** | **100,00%** | **0,00%** |

For people that need more details about lower-level tasks, go into the "Progress – By type" tab. It groups tasks by type, which you can then filter by whichever statuses that mean "In Progress" to you.



| Count of Status | Column Labels |
|---|---|
| Row Labels | In Progress |
| ⊟ **Bug** | **1** |
| Infinite scroll not loading after second data screen | 1 |
| ⊟ **Story** | **2** |
| Nested comment thread view | 1 |
| New user welcome message | 1 |

You can now easily filter to select the currently ongoing Epics, add lower level tasks if you need to, and paste them into an email or Slack message that you share with your stakeholders. With some tweaks, it's easy to reproduce the format of the template that we started from.



To: phoebe

Cc: tom

Bcc:

Subject: Work in Progress for Product X

Signature: None

Hi team,

Here's the status of what we're currently working on for Product X. There were some unexpected technical problems that delayed us for a bit, but we believe they're behind us and we'll be able to make up the time.

- **Gallery view (web)** (25% in progress, 75% still to do)
- **Lifecycle emails and notifications** (13% in progress and 87% still to do)
- **Nested comments** (25% released, 25% done but not released, 25% in progress and 25% still to do)

This is the detail of the tasks that are ongoing:

**Bugs**

- Infinite scroll not loading after second data screen

**Stories**

- Nested comment thread view
- New user welcome message

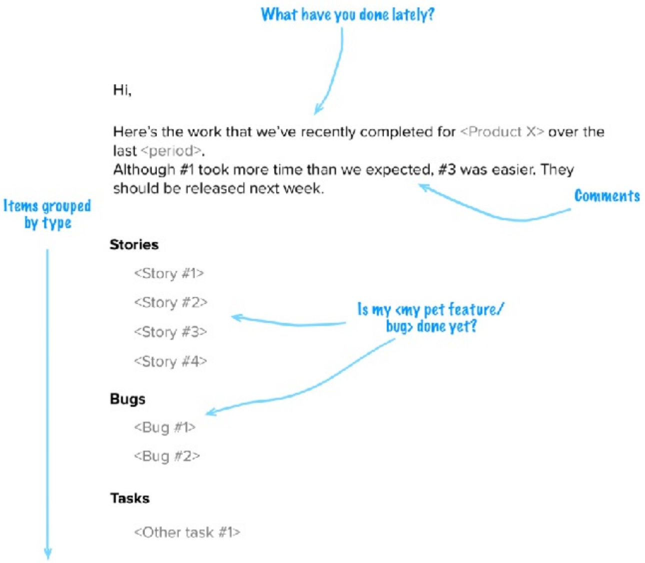
Best,
Daniel

**Pro-tip**: you can set up a personal subscription to the JIRA filter, so that you get its results emailed to you (as described here). You'll still need to go into JIRA to get the CSV version, but it works as a scheduled reminder for you to send the report out

(You could also subscribe a group of JIRA users and have them receive the search results themselves, but as you'll see, the format is just terrible for communicating with non-technical people and it requires that all of them have JIRA licenses.)

### Recently completed work

Sometimes, we just need to report back on what we've been up to. We need to inform what happened over a past period and add more context to explain if/why we over/underperformed (according to previous plans).


What have you done lately?
Comments
Items grouped by type
Is my <my pet feature/ bug> done yet?

Hi,

Here's the work that we've recently completed for <Product X> over the last <period>.
Although #1 took more time than we expected, #3 was easier. They should be released next week.

**Stories**

<Story #1>
<Story #2>
<Story #3>
<Story #4>

**Bugs**

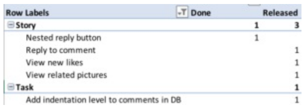
<Bug #1>
<Bug #2>

**Tasks**

<Other task #1>

#### How to create this with JIRA and Excel

The base for this report is in the “Recently completed” tab. Select the issue types that represent “Done” for you, and filter by resolution date, showing only what’s been resolved after the last time you sent the report. This assumes that your JIRA configuration sets a Resolution date on the issues that are ‘done’. You could also filter by Last Updated date for all your ‘done’ issues.

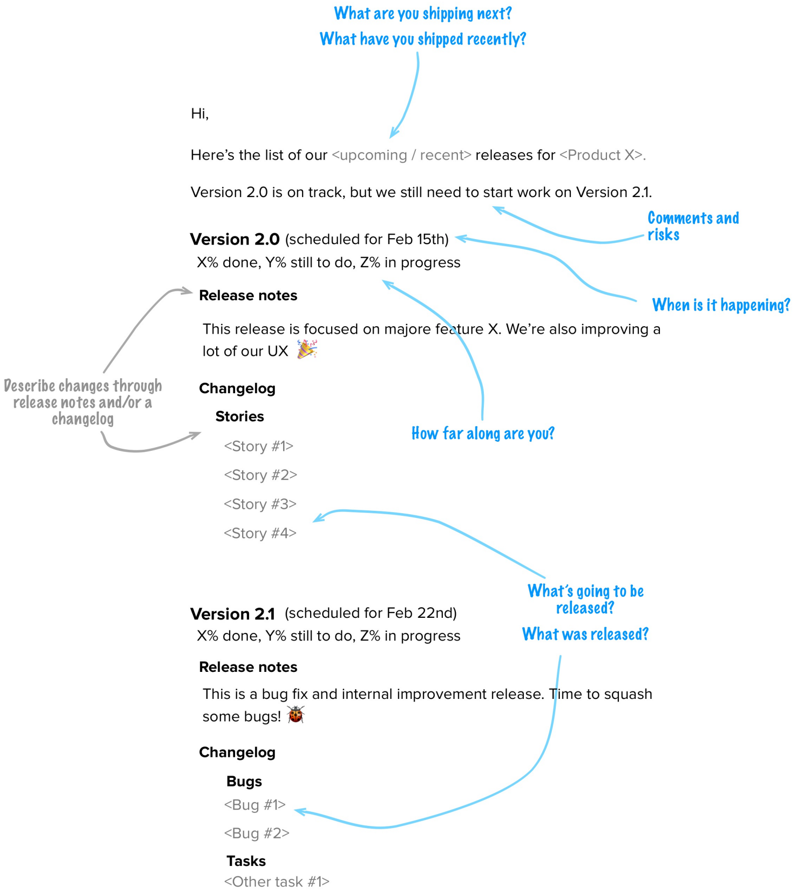


| Row Labels | Done | Released |
|---|---|---|
| ⊟ **Story** | **1** | **3** |
| Nested reply button | 1 | |
| Reply to comment | | 1 |
| View new likes | | 1 |
| View related pictures | | 1 |
| ⊟ **Task** | | **1** |
| Add indentation level to comments in DB | | 1 |

## Releases

### Recent and Upcoming

The template for recent and upcoming releases is essentially the same, but what changes is the point of reference. Some people need to know what’s coming up or as soon as something was released, while others only need to stay tuned to “what’s been going on”. This template addresses all of the most common questions around your releases.



What are you shipping next?
What have you shipped recently?

Hi,

Here’s the list of our <upcoming / recent> releases for <Product X>.

Version 2.0 is on track, but we still need to start work on Version 2.1.

**Version 2.0** (scheduled for Feb 15th)
X% done, Y% still to do, Z% in progress

**Release notes**

This release is focused on majore feature X. We’re also improving a lot of our UX 🎉

**Changelog**

**Stories**

<Story #1>

<Story #2>

<Story #3>

<Story #4>

**Version 2.1** (scheduled for Feb 22nd)
X% done, Y% still to do, Z% in progress

**Release notes**

This is a bug fix and internal improvement release. Time to squash some bugs! 🐞

**Changelog**

**Bugs**

<Bug #1>

<Bug #2>

**Tasks**

<Other task #1>

Comments and risks

When is it happening?

Describe changes through release notes and/or a changelog

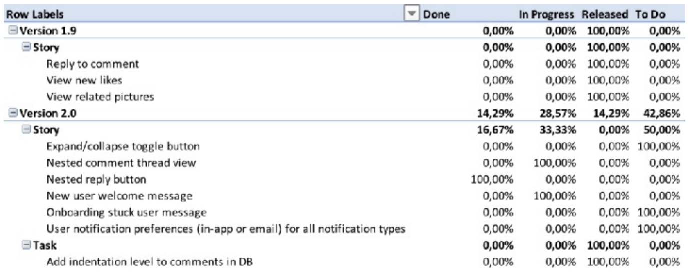
How far along are you?

What’s going to be released?
What was released?

Note that the “changes” section of the template refers to “release notes” and/or the “changelog”. Ideally, what you share with your stakeholders are easy to read, non-technical and engaging release notes with what most care about: the main changes and benefits from this release. The full list of issues that were part of the release (the changelog) are useful to certain stakeholders that need those types of details. Since the process below already produces a changelog grouped by issue type, to make it more readable, you should now have the time to create proper release notes

### How to create this with JIRA and Excel

Getting this out of JIRA is not as simple. Ideally, we’d be able to get the releases’ dates from the same issue search we did before, but that’s not possible, as far as I know. I haven’t found a better way to do this, so there’s a bit more manual work when creating these reports (but if you use the tool I’m building, it would all be automated). Anyways, if you go into the “Reports” tab in the spreadsheet, you’ll get a list of all releases that are linked with issues in JIRA.



| Row Labels | Done | In Progress | Released | To Do |
|---|---|---|---|---|
| ⊟ **Version 1.9** | **0,00%** | **0,00%** | **100,00%** | **0,00%** |
| ⊟ **Story** | **0,00%** | **0,00%** | **100,00%** | **0,00%** |
| Reply to comment | 0,00% | 0,00% | 100,00% | 0,00% |
| View new likes | 0,00% | 0,00% | 100,00% | 0,00% |
| View related pictures | 0,00% | 0,00% | 100,00% | 0,00% |
| ⊟ **Version 2.0** | **14,29%** | **28,57%** | **14,29%** | **42,86%** |
| ⊟ **Story** | **16,67%** | **33,33%** | **0,00%** | **50,00%** |
| Expand/collapse toggle button | 0,00% | 0,00% | 0,00% | 100,00% |
| Nested comment thread view | 0,00% | 100,00% | 0,00% | 0,00% |
| Nested reply button | 100,00% | 0,00% | 0,00% | 0,00% |
| New user welcome message | 0,00% | 100,00% | 0,00% | 0,00% |
| Onboarding stuck user message | 0,00% | 0,00% | 0,00% | 100,00% |
| User notification preferences (in-app or email) for all notification types | 0,00% | 0,00% | 0,00% | 100,00% |
| ⊟ **Task** | **0,00%** | **0,00%** | **100,00%** | **0,00%** |
| Add indentation level to comments in DB | 0,00% | 0,00% | 100,00% | 0,00% |

This lets us group them by release and issue type, creating a neat changelog. You could also get this from JIRA’s “Changelog” feature in the release page, but this way is faster and it comes from the same file you’re using for the other reports.

From this list, you would then need to choose the releases that you want people to know about (either upcoming or recent) and manually add their date (by looking at JIRA or your own memory).

## A step in the right direction

These templates and instructions should hopefully get you out of some status meetings and save you time. But they still take some manual effort and there are also a bunch of tiny little caveats due to how JIRA works and how your own instance may be set up. It’s a step in right direction, but it’s not enough. **Making sure that everyone’s updated about the day-to-day of the product is critical, but all the busywork and time we spend on it shouldn’t be.**

This is why I’m **creating a tool designed specifically for Product Managers to streamline and automate many of their internal product communications**. More information is coming soon and an early access program will open first to subscribers, so if you’re interested sign up below. Your feedback and support during this phase will let us improve the tool to better serve you; in return, you get to start saving time and a big discount.

Thanks for subscribing! Check your inbox for a welcome email. I'll also be in touch when the tool launches!

***

To read this article online with embedded hypertext links, go here:

**https://foldingburritos.com/articles/2018/02/13/report-templates-less-status-meetings/**

# About Daniel Zacarias

**Daniel**Zacarias is the founder of Substantive, a digital Product Management consultancy helping teams improve their strategy, discovery and delivery processes.

He’s fascinated by the human side of making software products. On the one hand, getting to understand customers’ problems and needs. On the other, dealing with all the internal stakeholders, teams, constraints and decisions that need to happen in order to get the product out the door. You can learn more on his blog or his company’s website.

# Nominating Committee

## About Saif Ahmed

**Saif** Ahmed is a Certified Scrum Professional. He is also a member of CTC special interest group and is actively pursuing Certified Team Coach role from Scrum Alliance. He has over 20 years of experience in IT and Business transformation. Always learning and discovering new ways of work. He has extensive experience in **large scale agile transformations**. He has trained and coached many individuals, teams, and organizations on lean values and agile principles. He has led and mentored global teams for application delivery and support. Currently working as an agile transformation coach at Bank of America. Saif Ahmed can be reached at safahmed@gmail.com

Linkedin: https://www.linkedin.com/in/saifuahmed/

## About Raj Chandler

**Raj** is a Certified Scrum Professional (CSP) and is actively pursuing the Certified Team Coach qualification from the Scrum Alliance. For over 15 years, Raj has supported large enterprises on transforming the delivery complex enterprise solutions from traditional project management to Agile. Recently Raj has been focusing on non-technical teams embrace and adopt Agile values, principles, and practices in order to delivery timely value. Raj is passionate about developing high-performing teams based on commitment, accountability, communication, trust, and fun. Raj currently works as an Agile Coach leading a journey of a large enterprise through an Agile Transformation.

Raj can be reached at rajeevchander@gmail.com or via linkedin: https://www.linkedin.com/in/rajchander/

## About Laisa de Almeida

**Laisa** de Almeida is an experienced servant leader currently working with Scrum and Kanban teams in Tampa, Florida. She helps organizations, teams and individuals on their journey to Agility. She is experienced in leading cross-functional, distributed and multicultural teams and is appreciated for her training skills and hands-on approach. Laisa has an undergraduate in Accounting and an MBA. She started her career in technology as a Business Analyst in the waterfall world. In 2012, she was introduced to Scrum and the Agile community as a Product Owner. From then on, she has been an enthusiastic adept of Agile. She is comfortable coaching in house teams in a consulting environment, both on-site and remotely AgStar, Pearson, IDEXX, Planet Fitness, Graham Media Broadcasting, Thomson Reuters, UHG WellMed, Citibank and PwC are among the companies she has worked with in a consulting capacity. Laisa enjoys staying in touch with the Agile community and participated in numerous conferences and user group events as volunteer, attendee and speaker. In her spare time, she enjoys the beach, music and travel. Chocolate, tea and margaritas are her favorites.

## About Michael de la Maza

**Michael** is a Scrum Alliance Certified Enterprise Coach (CEC). As an Agile consultant, his major engagements have been with Paypal, State Street, edX, Carbonite, Unum, and Symantec. He is the co-editor of *Agile Coaching: Wisdom from Practitioners and co-author of Why Agile Works: The Values Behind The Results*. He holds a PhD in Computer Science from MIT. He can be reached at michael.delamaza@gmail.com and his website is hearthealthyscrum.com.

## About Paulo Dias

**Paulo** Dias (CSP-SM, CSP-PO) is a passionate agile practitioner on the journey to become a CTC/CEC by the Scrum Alliance and an ACC/PCC by the International Coach Federation. He has 23 years' experience in Software Development having started his career as a software engineer in 1995. In the last 10 years, Paulo has been working on a variety of different roles as Enterprise Agility Coach, Consultant, Trainer and Mentor. As a coach, Paulo has been partnering with clients to help them bridge the gap between where they are and where they want to be, using skills such as active listening, asking transformational questions, empathizing and motivating clients. As a leadership consultant, Paulo has been helping organizations to gain strategic alignment, becoming more adaptive to fast changing market conditions, reducing lead time and increasing flow efficiency. The bulk of his experience has been in large financial institutions, having also worked in the airlines, telecommunications and automobile industries.
Paulo can be reached at: paulorcd@gmail.com and his linkedin profile is:
https://www.linkedin.com/in/paulo-dias-uk/

## About Alexander (Sasha) Frumkin

**Alexander** (Sasha) Frumkin (CSP-SM, CAL-1) is a passionate Agile Coach and trainer. He wrote his first program in Algol-60 at high school and still writes code for fun. Sasha spent 18 years of his professional life at Autodesk growing from a junior developer to a system architect and team lead. Later he led software development teams in multiple engineering companies in solar industry and co-founded a solar software start up.
Sasha discovered Agile Manifesto in 2007 and lives and breathes agile and scrum ever since. He believes that Psychological safety is a foundation for a House of Scrum.
Sasha partners with his clients in creating a culture of brutal transparency and experimentation, a culture of learning by doing.
Currently Sasha is a Principal Agile Coach at Bank of the West. He is on his journey to CTC/CST certifications with Scrum Alliance. He also facilitates Agile Practitioner On-line Special Interest Group.
Sasha can be reached at frumkia@yahoo.com and his profile is https://www.linkedin.com/in/alexander-frumkin-8770141/

# About Gary Hansen

**Gary** Hansen has worked with agile methods since 2009. His most notable clients have been Target, 3M, Best Buy, Thrivent Financial, and TCF Bank. He works with all levels of an enterprise to ensure that transformation efforts are aligned. Gary is an avid learner who continually looks for new ideas that challenge his thoughts. He has 13 active certifications and is pursuing a Scrum Alliance Certified Team Coach certification and an International Coaching Federation Associate Coaching Certification. Gary has an M.Ed. in Human Resource Development from the University of Minnesota. He can be reached at hansengaryw@gmail.com.

# About Chester Jackson

**Chester** Jackson is an Agile Coach and ICF Professional Certified Coach (PCC) with over 6 years of experience teaching, mentoring, and coaching teams and individuals at different levels of the organization. He is passionate about moving others through awareness into new action to help them make their mark in the world. His current focus is coaching groups to help them work together more effectively and growing the coaching skills of others around him to increase the impact that coaching can have in an organization.

Chester can be reached at chester.lynn.jackson@gmail.com and his LinkedIn profile is www.linkedin.com/in/chester-jackson

# About Oleksii Khodakivskyi

**Oleksii** Khodakivskyi helps companies to be competitive in the current changing world by enabling effective product development through collaboration.

He is a Scrum Alliance Certified Scrum Professional® (CSP-SM, CSP-PO) and Large-Scale Scrum (LeSS) Practitioner, comes with a degree in Information Technology Engineering and has more than 15 years of experience working in the IT industry.

As an Agile coach and trainer in Paris, France, he enjoys supporting organisations, departments and programs in the adoption of the Agile way of working and Lean thinking.

Oleksii can be reached at oleksii.khodakivskyi@gmail.comand his LinkedIn profile: https://www.linkedin.com/in/khodakivskyi-oleksii/

# About Matt Kirilov

**Matey (Matt)**’s passions lie in the interplay of disciplines that surround Agile and his interests deepen in both the professional/systems coaching and the teaming (team development) competencies. He is the organizer or co-organizer of several Agile and coaching meetups in Omaha (U.S.) and in Bulgaria. Matt currently is an Enterprise

Agile Coach and is the co-creator of the *From Scrum Master to Coach* training series. He runs a low-cost publishing house for agile and leand e-books. You can connect with Matt at mattdkirilov@gmail.com

## About Andrew Lin

**Andrew** has held different roles in agile as the Dev. Team member, the Scrum Master, the Product Owner, and Agile Coach since 2008, he now enjoys being a coach helping himself, organizations in Taiwan, and teams in the U.S. to transform.

The highlight of his agile journey so far is that he had a one-on-one interview with Jeff Sutherland in 2017. His mission is to be the agile bridge between Taiwan and the U.S. Andrew has built a platform www.agilegrandmaster.com that translate agile master's articles includes several versions of Scrum Guide, he believes everyone is a grandmaster, he hosts Scrum Alliance and Agile Alliance's group in Taiwan.

He lives in Fairfax, VA, you can reach Andrew at https://www.linkedin.com/in/andrewlinagile/

## About Christine (Christie) Murray

**Christie** Murray as an Enterprise Agile Coach focuses on helping people transform their organizational culture, mindset and practices to gain sustainable and continual growth in their Agile Journey.

Her trademarked Agile PS Culture and Communication Framework guides in creating PS (Psychological Safe) environments of Trust so that the benefits of Agile practices can be realized.

Her work takes her all over the world assisting Fortune 500 companies such as CISCO, American Express, Micron and ExxonMobil with their Agile Transformations.

She gives back to the Agile community as a volunteer providing coaching and mentoring to rising Agilists all over the world.

When she is not coaching or giving back to the Agile community, she is dedicated to helping shape DataOps, the next breakthrough innovation that fosters Data Agility into the Agile EcoSystem.

Feel free to reach out to her directly at cmurray@enterprisedataagility.com and to learn more about Data Agility and the DataOps Manifesto visit https://dataopscoachinginstitute.com/

## About Pelumi Olajide

**Pelumi** is an agile coach, facilitator and trainer with 10 years of experience working with teams utilizing agile methodologies for implementing Information technology solutions. She is a certified scrum professional (CSP), certified product owner (CSPO), certified agile leader (CAL1) and a certified Back of the room trainer. She is a certified Project Management Professional (PMP), certified ICAgile professional coach and facilitator (ICP-ACC, ICP –ACF).
Pelumi coaches agile teams, product owners and executives through continuous improvement. Pelumi is very passionate about helping individuals and teams be self organized, continuously improve and she enjoys partnering with her teams and clients in solving complex business issues through effective communication, analytical rigor and quantitative techniques. Her professional experience involves cross-functional teams and multi-location projects in North America, Europe, and Africa.
www.growinmindllc.com
https://www.linkedin.com/in/pmppelumi/

## About Allison Pollard

**Allison** Pollard helps people discover their agile instincts and develop their coaching abilities. As an agile coach with Improving in Dallas, Allison enjoys enabling others to become trusted leaders for change in their organization. In her experience, applying agile methods improves delivery, strengthens relationships, and builds trust between business and IT. Allison is also a Certified Professional Co-Active Coach, a foodie, and proud glasses wearer. Visit her blog at www.allisonpollard.com to connect further.

## About Albert Arul Prakash

**Albert** Arul Prakash is an Agile coach and an upcoming author working with teams to implement agile values and principles in their daily software development activities. He is also a senior engineering manager working with Mindtree implementing different engineering practices.

Albert is a certified scrum professional (CSP -SM, CSP- PO), certified product owner (CSPO), certified scrum master (CSM). As part of is engagement, he coaches agile teams, product owners and scrum master across the organization. Albert is very passionate about helping teams to continuously improve and be one step better than yesterday.

http://www.linkedin.com/in/albertarulprakash

## About Mohammed Rowther

**Mohammed** Rowther based in the United Kingdom is a **Scrum Alliance Certified Scrum Professional (CSP), Certified Agile Leader 1** and **Scrum@Scale Practitioner**. He comes with a degree in Information Technology Engineering and has more than 17 years of experience in the IT industry. He is currently working as an Agile Consultant/ Coach for various clients across the UK through MSquare Elite & Efficient Agile and has added immense value to each of the Organisation. He is a highly motivated and competent Agile practitioner and one of my key qualities is staying **Humble** & let go of the **Ego**. Patience is the key to Success in life and is open to **listening** to everyone whether we agree or disagree with what they say. **Respect** everyone around you and keep your mind clear. I believe in giving more than 100% in whatever you do and loves acknowledging the effort. I love cooking and play cricket. You can reach me on LinkedIn LinkedIn https://www.linkedin.com/in/mohammedrowther/ or www.msquare-elite.com.

## About Cherie Silas

**Cherie** is a Scrum Alliance, Certified Enterprise Coach (CEC) and an ICF Professional Certified Coach (PCC). She has a strong desire to help people arrive at the place they define as success in both personal and professional life. Her goal is to invest the experience and talents she has gathered through years of learning, often times the hard way, into people whom she hopes will become greater than she can ever dream to be.

Her main focus is culture transformation and development of people in Agile roles. When not coaching agile organizations, Cherie can be found coaching individuals, executives, employees, and rising leaders of non-profit organizations. She also teaches professional coaching to people in Agile careers who want to earn ICF credentials and add coaching to their toolkit. Cherie has background training from CTI and ORSC and has a passion to bringing professional coaching into the Agile world.

Cherie’s life mission that drives every interaction with every individual she encounters is simply this: To leave you better than I found you with each encounter

https://tandemcoachingllc.com

## About Sourav Singla

**Sourav** Singla, an enterprise Agile Ninja Coach with TCS, leveraged agile practices by exploiting the fundamentals of Agility for fortune 500 companies enabling continuous delivery of the highest value. He is part of 1000 Strong TCS Agile Coach Community that is part of TCS Enterprise Agile vision.

He is a Certified Team Coach (CTC).

He has guided many large enterprises through agile transformations and helped them on their journey to continuous improvement in their product development capability by working closely with teams, programs & leadership, understanding their potential, challenges & strategic goals.

Sourav is passionate about helping teams to deliver immediate value and exceed expectations through building high-performing teams and implementing proven agile methods.

As a coach he seeks clarity around purpose, shared vision & values to bring alignment towards achievement, innovation, and sustainability.

He gives importance to people side of change & focus on bringing meaningful change into organizations that can improve the lives of people.

Possess deep experience in agile practices, leadership coaching, organizational evolution & scaling agile.

He is a strong believer of distributed agile model & has good experience in creating & leading agile teams in distributed setup.

He lives in Mumbai, India, you can reach Sourav on LinkedIn (https://www.linkedin.com/in/sourav-singla-safe-agilist-csp-icp-acc-csm-cspo-ssm-lssg-35b65924/)

## About Ashwani Kumar Sinha

**Ashwani** is an avid Agile Practitioner with a growth mindset and an astute problem solver. Competent Project Management Professional with 10+ years of experience managing multi-year, multi-million dollar projects with a special focus on Risk Management & Quality Control. Successfully planned and delivered end-to-end cloud based technology projects in Digital Marketing, E-Commerce, Healthcare, Real Estate & Legal domain through multiple cross functional teams globally.

Ashwani can be reached at ashwani_sinha84@yahoo.com. His LinkedIn profile is https://www.linkedin.com/in/ashwani-kumar-sinha/

## About Ted Wallace

**Ted** Wallace is a Certified Scrum Professional (CSP) from the Scrum Alliance who is in the process of working towards being a Certified Team Coach (CTC). Ted has worked in both startup and corporate environments in developing their organizational and technical structures to be able to incorporate the Agile mindset. He loves seeing people, teams, and companies grow and gain the benefits of increased Agility. Ted holds masters degrees in Computer Science and Physiology from Maharishi University of Management.

Ted can be reached at tedtalktoday@gmail.com. His LinkedIn profile is https://www.linkedin.com/in/ted-wallace-9299757. And his blog is www.Iloveagilelife.com.

***

# Co-Editors

## About Cherie Silas

**Cherie** is a Scrum Alliance, Certified Enterprise Coach (CEC) and an ICF Professional Certified Coach (PCC). She has a strong desire to help people arrive at the place they define as success in both personal and professional life. Her goal is to invest the experience and talents she has gathered through years of learning, often times the hard way, into people whom she hopes will become greater than she can ever dream to be.

Her main focus is culture transformation and development of people in Agile roles. When not coaching agile organizations, Cherie can be found coaching individuals, executives, employees, and rising leaders of non-profit organizations. She also teaches professional coaching to people in Agile careers who want to earn ICF credentials and add coaching to their toolkit. Cherie has background training from CTI and ORSC and has a passion to bringing professional coaching into the Agile world.

Cherie's life mission that drives every interaction with every individual she encounters is simply this: To leave you better than I found you with each encounter

https://tandemcoachingllc.com

## About Michael de la Maza

**Michael** is a Scrum Alliance Certified Enterprise Coach (CEC). As an Agile consultant, his major engagements have been with Paypal, State Street, edX, Carbonite, Unum, and Symantec. He is the co-editor of *Agile Coaching: Wisdom from Practitioners and co-author of Why Agile Works: The Values Behind The Results*. He holds a PhD in Computer Science from MIT. He can be reached at michael.delamaza@gmail.com and his website is hearthealthyscrum.com.

***

# Table of Contents