BEST AGILE ARTICLES 2020

Michael de la Maza, CEC • Cherie Silas, MCC

Best Agile Articles 2020

Michael de la Maza, CEC • **Cherie Silas,** MCC

BEST AGILE ARTICLES 2020

Michael de la Maza, CEC ● Cherie Silas, MCC

Printed in the United States of America

ISBN: 9798718925111

Design and Layout: Alex Kudinov
Fourth Edition: April 2021

Disclaimer: *The views, opinions, and pictorial assumptions expressed in the articles contained in this publication are those of the article authors and do not necessarily reflect the views of the publication curators, editors, volunteers, sponsors, or members of the agile community involved in the publication.*

Scrum.org™

The Home of Scrum

Professional Scrum Training

- APPLYING APS PROFESSIONAL SCRUM
- APPLYING APS SOFTWARE PROFESSIONAL SCRUM
- PROFESSIONAL SCRUM PSM MASTER
- PROFESSIONAL SCRUM PSM II MASTER
- PROFESSIONAL AGILE PAL-E LEADERSHIP ESSENTIALS
- PROFESSIONAL SCRUM PSPO PRODUCT OWNER
- PROFESSIONAL SCRUM PSPO ADVANCED PRODUCT OWNER
- SCALED SPS PROFESSIONAL SCRUM
- PROFESSIONAL SCRUM PSK WITH KANBAN
- PROFESSIONAL SCRUM PSU WITH UX

Professional Scrum Certifications

- PSM I
- PSM II
- PSM III
- PSPO I
- PSPO II
- PSPO III
- PSD I
- SPS
- PAL I
- PAL EBM
- PSK I
- PSU I

Table of Contents

Foreword

We are delighted to bring you this volume of the best agile articles of 2020. Our goal in publishing this book is 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 Review Committee of twenty-four people with expertise in a variety of areas related to agile.

- The agile community. A call for nominations went out in early 2020 and over 120 articles were nominated by the community. We selected the top 50 articles to present in the publication.

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 fourth book in the series. Previous books, Best Agile Articles of 2017, 2018, and 2019, are available on Amazon and on the website at *https://baa.tco.ac/books*.

We are thankful for the great participation of 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, MCC
cherie@tandemcoaching.academy
Dallas, TX USA

Best Agile Articles 2020
View this publication online or download for free at
https://baa.tco.ac/2020


Tandem Coaching
ACADEMY
DISCOVER
COACHING©
START YOUR COACHING JOURNEY
TCO.AC/DISCOVER

Editor's Choice

Articles in this section were deemed
the "Best of the Best" Agile articles of 2020.
They received strong recommendations from
the selection committee and have been deemed
by the editors as "Must Publish" writings
worthy of special recognition.

Shu Ha Ri of Professional Coaching

By Alex Kudinov, Erica J Henson

Shu-Ha-Ri has been a frequent topic within Agile communities for years to capture the essence, the progress of the agile journey we embark on as we grow and learn better ways of working. Many Agile leaders have written about Shu-Ha-Ri, from Martin Fowler to Alistair Cockburn both creators of the Agile Manifesto. But what is it, and why does it matter?

To understand what Shu-Ha-Ri is in Agile, it is essential first to know its origin. The concept of Shu-Ha-Ri originates from the Japanese martial art Aikido. Aikido master End Seishir Shihan explains the idea as follows:

It is known that, when we learn or train in something, we pass through the stages of Shu, Ha, and Ri. These stages are explained as follows. In Shu, we repeat the forms and discipline ourselves so that our bodies absorb the forms that our forebears created. We remain faithful to these forms with no deviation. Next, in the stage of Ha, once we have disciplined ourselves to acquire the forms and movements, we make innovations. In this process, the forms may be broken and discarded. Finally, in Ri, we completely depart from the forms, open the door to creative technique, and arrive in a place where we act following what our heart/mind desires, unhindered while not overstepping laws.

Jeff Sutherland, the co-creator of Scrum, leased these concepts to describe the three states of Scrum Mastery and its progression. He rightly identified the same stages a Scrum Masters must pass through to achieve a deeper master of the practice.

Let us walk through a simple explanation of how it applies to Scrum Masters.

Shu

In the Shu state – the beginning stage, the Scrum Master is the master of the process, of the Scrum framework. She is good at going through motions of basic practices: setting up and facilitating events, helping her team achieve a stable state, helping the team find its velocity, and guiding continuous improvement through the process of retrospective. Her agile training finally pays off, however, in this state, the focus is more on how to achieve something without worrying too much about underlying theories of agile practice.

Ha

When the Scrum Master and her team achieve the Ha state, they are doing more than executing steps to follow a prescribed way of working. The Scrum Master with a better understanding of the underlying theory and principles feels more confident about her abilities, let's go of the agile techniques mechanistic structure. According to Jeff Sutherland, the team can get software done at the end of the Sprint and has a good Product owner with a ready backlog at the beginning of the Sprint, has data that clearly show at least a doubling of productivity, and has strong management support.

Ri

In this ultimate form, a Scrum Master can step back and let the team, a well-functioning and productive organism, perform at their best. The Scrum Master is never in the spotlight and is never a puppeteer. The Scrum Master instinctively and seamlessly adapts to the environment and team needs as frequently as needed. That's where the whole initiative of the agile transformation might start making sense.

Throughout my career, I have not seen a Scrum Master continuously operate within the team in a Ri state. I suspect there are a few reasons for that. First and foremost, the culprit could be the management. Probably for them, the Ha state is the Holy Grail of productivity. When management sees the teams achieve the Ha state, they could be easily enticed to spread success, and associated agile practices, across the organization. And boy, are they doing it wrong. The usual way the management goes about scaling Scrum in organizations is what we call a Copy/Paste Scrum. A Ha-level successful team gets disbanded to move the best players to different teams ignoring the team dynamics and resulting in the loss of velocity, camaraderie, and cohesiveness. The Scrum Master who helped the team achieve the Ha state goes her merry way, hopefully on to making another team great.

The second reason, I surmise, is that team development is not going in a straight upward only direction. Teams are living organisms. Organisms have good and bad days. They fall sick, and they heal. Teams are no different. They evolve, they improve, they get to the new heights, and they regress. Regression is painful and not tolerated well by a high-functioning team used to its high-performance.

Setbacks for such teams can be challenging to navigate. Ideally, a well-designed and functioning mechanism of inspection and adaptation should serve as a safety net for teams in such situations. When that safety net is present and robust, it catches the team and springs it back up again.

So how does a Scrum Master employ and remain in the elusive Ri state? Alistair Cockburn, in Heart of Agile , says that Ri-level people generally cannot say how they decide on a technique at the moment, because it is so ingrained and immediate. In general knowledge acquisition terms, Ri corresponds to invent and blend techniques. To borrow from Martin Broadwell's four stages of competence , the Ri state roughly corresponds to unconscious competence.

The main skills a Scrum Master employs in the Ri stage are those of professional coaching.

So how do great Scrum Masters acquire the true mastery of professional coaching skills? How do they hone and develop them? How they keep the momentum going and the continuous improvement chugging along? I would argue that the development of the professional coaching skills by Scrum Masters follows the same steps of Shu-Ha-Ri or the four stages of competence.

The International Coach Federation (ICF) defines coaching as a coach-client partnership in a thought-provoking and creative process that inspires clients to maximize their personal and professional potential. Apply that to a Scrum Master, and you will see that it is precisely what great Scrum Masters do. Here are some useful examples. They challenge their teams to greatness. They raise awareness and build accountability. When a new Scrum Master takes her first step from unconscious incompetence to conscious incompetence (Shu), the sheer amount of knowledge, skills, techniques, and mindset changes are overwhelming. That is where the concept and the mindset of continuous, incremental improvement come really handy.

The Shu state is not for the faint of heart. It is not for those who are looking for quick wins. It is a hard work of learning, practicing, inspecting, and adapting, it is an ultimate learning process. Unfortunately, coaching skills do not make it to the top of their priority list at this stage, and that is a shame. Here, at the Shu level, Scrum Masters are working just way too hard and step in way too many situations that they should not have. In many occurrences this state is compounded by the fact that a Scrum Master is assigned to a new agile team with merely a nascent teams process. What they do not realize is that solid professional coaching skills appropriately utilized would have made their lives much easier throughout their learning journeys, and most importantly, at this Shu state. The sooner Scrum Masters are introduced to professional coaching skills, the easier their path to real mastery will be.

There is a vast misconception about professional coaching skills. Some people believe that coaching is nothing more than sitting on the couch and asking powerful questions. And that belief has its roots. It stems from the subpar education some coaches have received combined with limited knowledge of tools, techniques, and competencies of professional coaching. There is no argument that asking powerful questions is an excellent tool in the toolset of every coach. But a tool is only a tool. It is not merely enough.

Coaching first and foremost is a mindset, one that genuinely believes that a client is naturally creative, resourceful, and whole and doesn't need to be fixed. A coach believes that a client has all the resources he needs to achieve his goals.

Coaching mindset is immensely curious. Curiosity is an integral part of the growth mindset the mindset that is characterized by an insatiable drive to learn, to push boundaries, and to improve continuously. That curiosity brings such words into a Scrum Master's lexicon as: what if, I wonder, I'm curious. Notice that none of these are directive or prescriptive. This mindset allows its owner to be open to possibilities, new opportunities, and enables the belief that the team has all the answers it needs to be great.

The Shu Coach

When a coach enters the *Shu* state, she is already aware of the necessity to possess and potentially transform her mindset to that of curiosity and wonder. She is aware of some essential competencies a professional coach holds dear. She has some necessary tools in her coaching toolbox. She relentlessly exploits these tools and builds her competences while staying fully aware of the fact that only *practice* and a lot of it allows for the building of those competencies. She speaks less and listens more. She becomes very comfortable with silence. She starts listening not to respond but to understand. She makes mistakes and a lot of them. And she becomes aware of those mistakes. And she practices, and practices, and practices some more. She joins a group of like-minded Scrum Masters to practice even more.

As time goes by, she finds less and less need to use a directive approach to tell her team what to do as they mature in their practices and grow in their cohesiveness. She finds that her daytime job turns into that earlier elusive partnership where her role is that of support, encouragement, and acknowledgment. The buildup of the coaching competencies allows for the emergence of a well-rounded, albeit still a novice and inexperienced coach.

The *Shu* state of professional coaching skills roughly corresponds to the ICF ACC accreditation. The latter requires some formal training and education, which makes sense as the ICF is committed to upholding high standards of coaching skills amongst its members. Can a Scrum Master build their professional coaching skill competence to the level of ACC without formal education? Absolutely. This certification, like any other competency-based one, is an acknowledgment and confirmation of the level of the coaching capabilities achieved.

How do coaches transition to the *Ha* state?

Coaches will transition into *Ha* by actively learning more, practicing harder, and continuously expanding their toolbox. At this level, active listening to the whole person and coaching a person rather than a problem is mandatory.

If you are a Scrum Master, this is the level where you start digging deeper into the essence of the team. You begin noticing and addressing the interpersonal relations, tensions, conflicts. You realize that the team is capable of solving their problems, and your primary role is to help them uncover and understand their beautiful inner selves.

You recognize that, while skills and experience are essential, the intricacies of the personal relationships and team dynamics are what move them forward on their road to greatness. You, as a coach, are highly skilled at bringing about your teams' self-awareness and building their accountability. You lovingly support the team while firmly and steadfastly challenging them. You become a master of balancing these two.

You realize that the effort you had to expend to practice all those coaching competencies at the *Shu* level diminish greatly at *Ha* level. You become *consciously competent*: you begin mastering the art of choosing the right tool for the right circumstances and do so seemingly. And you may be thinking of that coveted ICF PCC certification, which reflects the training, practice and experience, and adherence to the highest ethical standards coaching offers.

The Elusive Ri

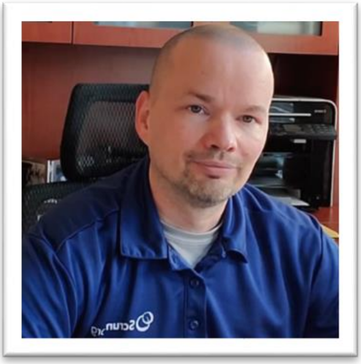
At the Ri level, you are not beholden to any framework, method, or ideology. You pick what feels right at the moment and use it with seamless mastery. Your skills and experience are well developed and so vast that every situation prompts the right response. Can you still be wrong? Yes, you can. But you inspect, adapt, and fix–never letting the same mistake happen again. You effortlessly incorporate the environment and ecology into your work and coaching practices. They send signals that evaded your attention at both *Shu* and *Ha* levels. You coach the team as a living, breathing, whole organism as it is while understanding its complex laws of interactions and behaviors. You master scaling the ladder of environment, behaviors, capabilities, values, beliefs, identity, and spirituality up and down and up again, knowing intuitively on which step of that ladder a current situation belongs. Your learnings never stop. The layers and layers of new knowledge now interact with existing skills, behaviors, and practices, allowing you to create your unique style and way of thinking. You are not bound by any coaching tools or techniques that you learned in the previous two stages. Your skills transcend those and allow you to create the practices that serve you best in your unique environment.

The Ri state is exceptionally coveted. There is no yellow brick road anyone can show you that will lead you to it. Just keep in mind that some things in common for those who have walked the *Ri* path are hard work, an open mind, and an insatiable curiosity. Your mission is to find your way there, and you must want it.

Are you ready to start your journey?

This article was originally published at *https://baa.tco.ac/32mn*

About Alex Kudinov



Alex is a technologist, Professional Coach, and Trainer. He delivers business competitive advantage to his customers by growing and nurturing great agile development organizations.

At Tandem Coaching Academy, where we keep agile coaching non-denominational, Alex is a Managing Director. We work with agilists of all walks, bringing agile and professional coaching closer together.

Alex holds the International Coach Federation Professional certified Coach (PCC) accreditation. He is also a Board Certified Coach, Certified Executive Coach, Professional Scrum Trainer with Scrum.org, Scrum Alliance Certified Enterprise and Team Coach (CEC and CTC) and Kanban University Accredited Kanban Trainer (AKT).

He holds an MBA degree from the University of Texas at Austin.

Ways to Contact Alex

tandemcoaching.academy

info@tandemcoaching.academy

www.linkedin.com/in/alexkudinov

twitter.com/AlexKudinovPST

About Erica J Henson

Erica loves helping people achieve success in software delivery and creating long-lasting relationships that result in outstanding value to customers. She specializes in coaching teams to empower them to become high-performing and self-organizing.

Erica has 25+ years of software engineering and team development experience, holds PSM2, KMP2, and PSPO certifications, and is in the process of acquiring her ICF coaching accreditation.

She is the proud mother of the world's greatest 8-year-old, loves coffee, singing, and is constantly looking for ways to fan the flames of transformation for Good.

Ways to Contact Erica

www.raylantech.com

erica@raylantech.com

linkedin.com/in/ericajanine

twitter.com/ericajanine

Business Outcome-Based Metrics: How to Effectively Measure Your Agile Transformation Journey

By Andy Cleff

Outputs or Outcomes?

Many organizations are stuck measuring and making decisions based on outputs – like velocity. In fact, team velocity is one of the most commonly used, abused, and misused metrics in Agile software development as well as during digital transformations. Teams, their managers, and even their stakeholders often focus on "improving velocity" without considering the entire value delivery system. Then they are shocked when they don't get the business outcomes they really want, for example predictability or speed.

In this article, we explore healthy ways for your organization to use metrics to gain meaningful insights into the results of your experiments in the course of your Agile transformation.

Measure for a Purpose

As Simon Sinek famously says: "It all starts with why." You need to understand what you think you are going to measure and why you want to measure it. When it comes to metrics in Agile, the data itself is not the goal — instead, it's a means of tracking your journey, testing hypotheses, and providing feedback as you head towards your next goal.

That goal, the Big Why, should be focused on business outcomes, not outputs. Here are nine business outcomes (that come from Agile Velocity's *Path to Agility®* framework) to think about:

1. **Employee Engagement**: Employees are more satisfied in their work, willing to go the extra mile, passionate about the purpose of their jobs, and committed to the organization.
2. **Continuous Improvement**: The ability of the organization to relentlessly pursue optimizations in all aspects of business functions.
3. **Innovation:** New ideas, creative thoughts, or novel imaginations provide better solutions to meet new requirements, unarticulated needs, or known market needs.
4. **Customer Satisfaction:** Customers are satisfied with the experience, benefits, and outcomes when using your product or service.
5. **Market Responsiveness:** The ability of the organization to pivot quickly to respond to ever-changing market demands.
6. **Productivity:** Teams maintain a predictable cadence of delivery enabling the business to make informed business decisions.
7. **Speed:** The time it takes to deliver an idea into the market.
8. **Quality:** The product or service meets the expectations

of the market for usability, reliability.

9. **Predictability:** Increase the business value realized while maintaining or reducing costs.

We recommend that your organization pick one. Sorry, you can't have them all at once! Limiting your organizational WIP (Work in Progress) helps create a clear sense of urgency.

Reasons To Measure

There are many reasons to measure as you track progress towards organizational business outcomes during your transformation. These include:

- Knowing where to invest your money – and to do so based on more than just gut instinct
- Knowing if you are building the right things for our market
- Measuring your performance and alignment — where your inventory (the software that the development teams create) that addresses features, defects, risks, and debt is not easily visible
- Knowing if your customers and employees are delighted (or not)

Why Don't More Organizations Measure More Things?

In our practice, we routinely discover that many organizations don't have much in place in terms of metrics. Why not? Some reason include:

- People are afraid of weaponized metrics – ones used not for purposes of continuous improvement, but instead for comparison and punishment.
- They were collecting vanity metrics, ones that didn't offer any predictive power and eventually considered measurement programs "useless."
- Metrics are actionable, however, they are presented in a

way that people don’t find them useful (for example endless tables of numbers instead of clear graphic representations).

- Folks believed the “right things” were just too expensive to measure, and they failed to poke around to find existing data that would suffice.
- They mistakenly thought the “right things” were immeasurable — and didn’t consider finding proxies.

Meaningful Agile Metrics for Digital Transformations

Successful organizational transformations put in place metrics programs that focus on global outcomes and meaningful feedback loops. This is true at any point along the transformation journey.

About Andy Cleff

Andy is an experienced and pragmatic agile practitioner that takes teams beyond getting agile to embracing agile. His superpowers include holding multiple perspectives, enabling step changes that bridge current and future states, allowing space for both order and chaos simultaneously, celebrating success, and foremost, putting the people at the top of the priority list.

His chief weapons are well-asked questions, insightful retrospectives, and an ability to withstand awkward silences. And if all else fails, beer.

Ways to Contact Andy

www.andycleff.com

andy.cleff@agilevelocity.com

www.linkedin.com/in/andycleff

twitter.com/justsitthere

Engaging and Being Engaged as a Professional Coach – Candid Advice for Coaches & Clients

By Gene Gendel

Key Takeaways

- There are things that coaches need to do prior to engaging with a client company
- Coaches need to know about challenges they might be facing while trying to engage with a company
- There are some practical things coaches and companies can do to set up the engagement for success
- Things that hiring companies need to know about a coaching profession
- Advice for hiring companies to not shut doors (intentionally or unintentionally) in front of experienced coaches, without opening up flood gates for charlatans, cheer-leaders, and 'best-practices" experts

Have you ever found yourself in a situation, where you feel that you have so much great value to bring to your potential client, that you are so much better than anyone else they may have considered for the role so far …. yet, a client is hesitant to bring you in? Why?

Below are some recommendations for professional coaches:

You Must Self-Assess:

- Are you a professional trainer, coach and organizational design consultant who had spent many years honing his/her craft and investing in self-development? Do you consider yourself a life-long learner who openly speaks about it?
- Over the years, have you earned some of the highest industry-recognized accreditation that represents your professional journey? Do you make it explicit and visible publicly (e.g. LinkedIn, your web site)?

- Have you, not only read dozens of books and case studies but also authored or co-authored some on your own? Do you blog, write articles about your experiences and beliefs?
- Do you speak publicly (webinars, conferences), have your own web site, where you offer free educational information to communities? Do you have tens of thousands of followers in the industry, attending your events, reading your newsletters and benefiting from your expertise?
- Are you a one-person entity that you have relentlessly built over the years? Are you responsible for your own networking and business development, along with providing community service (this almost always keeps you busy at nights and on weekends), ON TOP of your paid work?

If you answered 'yes' to most of the above, you are actually might be facing some potential challenges.

Understand, What Are Some of Your Potential Challenges?

- When trying to engage with a client, if you sound very knowledgeable (even if it comes so natural to you), you might be creating an inflated impression about what your aspirations and intentions are. Without any intention, you could be perceived as someone who wants to come in, take over, set your own tone, and run the show.
- If you attempt to impress others with your knowledge to soon, the effect could be the opposite to what you expect: you might be putting the people that you interact within an uncomfortable position, some of whom could be much less experienced than you, unable to speak at your level, and perhaps even aiming for your role in the future,

because their old roles have been being downsized (Larman's Law #4). This is not so much a problem with senior management but quite common with first-second level managers and single-function roles (e.g. BAs, manual testers) that are sometimes asked to validate/interview you initially.

- You might be competing against a very large population of external candidates-consultants that are usually procured through staffing agencies/traditional preferred vendors, and presented as "coaches" (today, coaching is a highly commoditized and heavily overloaded term). In reality, such candidates are coaches in-name-only (a.k.a. coaches-centaurs). They are willing to engage at a deeply discounted rate and are easily inserted into an existing reporting structure (seamlessly fit an existing staffing model) just like any other, traditional "human resource". Once in, they typically become individual performers (e.g. tool administrators, backlog stewards, or metrics collectors) – a classic coaching anti-pattern (read more about anti-patterns here).
- You might be trying to enter into a client company's domain that is tightly controlled by a large consultancy that has already brought in their own, very expensive resources, installed their own, home-baked 'best practices' that are presented in the form of heavy power point decks and play-books. Even if you have a lot of experience with such best practices, unless you explicitly express your strong support for them, at the time of initial contact with a client, you might be perceived as a challenger and considered unfit.
- Although the most generous rate for a highly experienced independent professional (we assume, you are reading this now) is just a fraction of what a large consultancy would be charging a client for each placed consultant, if you cost more than a coach-centaur (described above),

you may put yourself out of range.

Understand, What Can You Do to Mitigate These Challenges?

First and foremost, remember: there is only one chance to make a first impression!!! You are lucky if you will have a second chance.

- Try to understand, really well, what the client's real goals and aspirations are. When you meet a client, even for the first time (first interview or just an informal lunch meeting), listen to THEIR concerns and feel for THEIR pains. Keep your strong views to yourself and tone down what you may know: do not overwhelm the client. At times, a client will not be explicit with you about their real problems, so you may need to read between the lines, ask probing questions. But be careful, how deep you probe. Don't become obvious.
- Do not up-sell yourself. Do not speak about your own qualifications, credentials or past successes (unless it is absolutely necessary or you are being asked). Of course, it would be amazing if someone paved the way for you, spoke highly of you and talked about how you have helped other organizations. If you have to refer to your past experiences, present them as circumstantial and based on past conditions (try not to sound absolute and categorical).
- Always remember that when you meet a client, you are on THEIR territory, and eventually, you will leave and they will own everything you have done for them. You may not even have a chance to claim credit for your work because its results will be seen only after you are gone. Therefore, be very explicit about your intentions upfront, of NOT wanting to 'take over, become a hero,

challenge everyone and change everything'. This could be a tough one, because at times, against your own will, you could be viewed as a leader-challenger, due to your perceived seasoning and expertise. You must make the client comfortable that you will respect their territory and their decisions and you are there to serve THEM.

- During your initial interaction with a potential client, even if you discover something that makes you feel that an immediate course correction is required, refrain from stating this too soon (unless you are explicitly asked to provide your own view). It would be wiser to offer assurance to a client that you are seeing a lot of potential for working together and you are ready to support them in any of their efforts. Then, only after you fully engage and dig in, should you start gently steering the client in a different direction by, reflecting on what you see, and offering alternatives, as needed.

Recommendations for companies looking for professional coaches:

Let's face it, today, finding an experienced and credible agile coach, is not easy. If you disagree with this statement, you are either very lucky and have special access to some great talent (e.g. referrals or networking) OR your perception of the role may need to change.

There is no need to be ashamed of not being able to find a good coach. You are not alone; many companies face the same challenge.

Truth be told, unfortunately, the industry has changed significantly over the last few years and this has become the source of many problems (some classic problems are described *here*). Today, the term "Senior Agile Coach" has been grossly diluted.

But fortunately, there are still great standards and guidelines you can follow when looking for an agile coach, irrespective of industry trends. Please, consider the dimensions below when looking for a professional agile coach for your organization. The original sources of these requirements are listed at the bottom of this page and you are encouraged to explore them for additional details.

Please, do not reduce, simplify or trivialize some of the key expectations of a professional agile coach. Because, if you do, these two problems will inevitably follow:

- Industry coaching quality (average) will be further decreased, and even if you don't care about this fact as much…you will care about the next fact….
- Quality of service to your own organization will be also low

So… with that, what are the most important "Must-Have" for Professional Agile Coach?

Quantitative Assets

- Has significant hands-on experience in at least one of the roles on an agile team
- Has coached multiple organizations, departments, or programs
- Has, at least, 1000 hours of experience coaching at the enterprise/organizational level or a combination of enterprise and multi-team level coaching
- Has diversity of coaching experiences that can be demonstrated, using different client engagement examples, and which include experience at the organizational level

Demonstration of deep knowledge

- Has formal and informal education about coaching and strong mentor relationships
- Has good working knowledge of Agile and Lean values, principles, and practices.
- Has helped individuals, teams, and leadership to

understand and apply Agile and Lean values, principles, and practices effectively

- Understands the dynamics, patterns, and development of multi-level teams and how they interact at the organizational level
- Knows the difference between consulting and coaching and knows when to apply each.

Ability to clearly articulate and substantiate one's own

- Coaching Career Overview (coaching, agile history and how a person got where he/she is today. Include key milestone years)
- Coaching Focus (summary of a person's professional self today, including a coaching approach and/or philosophy to coaching)
- Coaching Goals (personal development goals in coaching)
- Formal Coaching Education (formal education activities which have contributed significantly to their coaching journey. This includes a wide range of courses on topics including facilitation, leadership, consulting, coaching, process, tools, techniques, frameworks, and other related activities which have influenced our coaching practice)
- Formal Mentor-ship Education (coach, mentor-ship and significant collaboration activities where a person has DEVELOPED a skill or technique or RECEIVED guidance to his/her coaching approach and mindset.)
- Informal Coaching Learning (significant topics you have studied outside of the agile literature which has impacted his/her coaching approach or coaching philosophy)
- Agile Community Participation (agile community events, such as user groups, gatherings, retreats, camps, conferences, etc. in which a coach has participated)
- Agile Community Leadership (leadership contributions to

the agile community (e.g. writing, publishing, presenting, facilitating, organizing, training and other activities) through events, publications, courses, blogs and forums)

- Agile Community Collaborative Mentoring & Advisory (significant collaborative agile mentoring, advisory activities, where a person was mentoring, advising other individuals to increase their competency or in development of a specific goal)
- Coaching Tools, Techniques or Frameworks known (coaching tools, techniques or frameworks which you have implemented, customized, co-developed or developed in one or more client engagements)

Skills, Tools & Techniques

- Has contributed to significant improvements in organizations or departments through coaching techniques
- Has helped organizations and teams beyond the basics of Scrum theory and practice
- Has enabled organizations to find their own solutions to business problems through the application of Agile principles
- Is familiar with, promotes and embodies the mindset of Servant Leadership
- Uses a rich set of facilitation, training and coaching tools, and models

Personal Qualities

- Coaching Mindset Coaching skills/practices and frameworks
- Evidence that the coach has taken both their Experience and Learning and synthesized these into definitive practices, frameworks, approaches, and strategies)
- Self-awareness: Able to reflect on their own contribution

to the coaching by virtue of their own 'being'

- Constant Learning: Has and continues to acquire Coaching oriented learning through multiple dimensions
- Diversity of Experience with different types & sizes of organizations
- Participation in the Agile community

Conclusion

To summarize, being a highly qualified and experienced professional does not automatically qualify you as the best candidate to be selected. There are many situational conditions that must be considered while interacting with a potential client. Often times, not all your assets and resources should be revealed at once and to everyone. You may have to be strategically smart in your pursuit and goals, even if your intentions are most genuine and it feels that you just have to be yourself and hold nothing back.

But the flip side is true too. A company must have its own qualified people (not necessarily coaches in the title) that are able to identify and validate real, experienced, professional coaches that come from outside. Otherwise, they will likely find themselves being part of a vicious downward spiral; further deteriorating coaching quality and effectiveness, described in more detail here (*You Get What you Ask For: Agile Coaches- "Centaurs"*), and harming itself along the way.

This article was originally published at
https://baa.tco.ac/34ln

About Gene Gendel

Gene Gendel is an organizational design specialist, agile/lean coach and trainer, consultant and advisor to senior leadership with 20+ years of experience. Gene is dedicated to working with companies of various sizes and lines of business, trying to help them improve internal dynamics, organizational structure and becoming a better place to work.

Gene engages at all organizational levels: senior- and mid-level management, teams and individuals. In his work, Gene uses various methods, tools and techniques to amplify learning by others and to ensure that people gain autonomy after Gene "coaches himself out of the job".

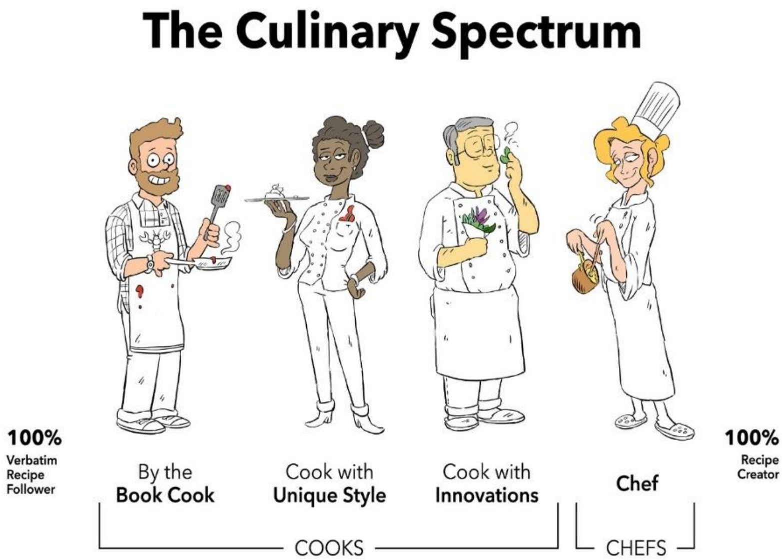
Ways to Contact Gene

www.keystepstosuccess.com/about-me

gg_nyc@yahoo.com

Cooks and Chefs: Why Agile Has not Fixed Our Problems

By Miljan Bajic



If you haven't had an "Agile" evangelist (like myself) hunt you down yet, here's what to expect. We might suggest that if you don't know how Agile and one of its popular frameworks will revolutionize your product, service, or industry, we think your company is squarely located under a rock.

Okay, I'm just kidding. But you know what I'm saying? Agile has become a jungle of tooling vendors and consulting companies selling frameworks that are implemented as a blueprint. There are too many cooks, when what we need are chefs. Let me explain…

Cooks follow recipes. Chefs create them. It's much easier to copy recipes that work, and for many aspects of our life, it probably makes sense to act as a cook. Cooks span a wide range.

Cook by the Book

On one end, we have cooks who only cook by following a recipe to the letter, carefully measuring every ingredient precisely the way the recipe dictates. The result is a delicious meal that tastes exactly the way the recipe has it designed.

In the professional consulting and coaching industry, this is somebody who swears by their favorite framework and acts dogmatically like they just drank the agile kool-aid. How many people like this do you know?

Cook with Unique Style

Down the range a bit, we have more of a confident cook — someone with experience who gets the general gist of the recipe and then uses her skills and instincts to do it her way. The result is something a little more unique to her style that tastes like the recipe but not quite.

The professional consulting and coaching industry is full of cooks with a unique style. Everybody today calls themselves an agile coach and consultant. They all tend to claim that they have a unique style and experience. All they have is the general gist of agile frameworks and some confidence at this point in the spectrum.

Cook with Innovations

At the far end of the cook range, we have an innovator who makes her version. But what all of these cooks have in common is their starting point is something that already exists. Even the innovative cook is still making an iteration of a burger, a pizza, and a cake.

In the professional consulting and coaching industry, these are the experienced coaches and consultants that understand the underlying patterns and principles behind the frameworks. For cooks, even the more innovative kind, there's almost always a ceiling on the size of the splash they can make in the world unless there's some luck involved. In the past, there was nothing wrong with being a cook. In today's environment, cooks without chefs to help them grow will not help you push boundaries and create solutions to the disruptions your industry faces presently and in the future.

Chef

At the very end of the spectrum, we have the chef. The chef reasons from principles, and for the chef, the principles are the available raw edible ingredients. Those are her puzzle pieces, her building blocks, and she works her way upwards from there, using what's available, her experience, her instincts, and her taste buds. The chef creates while the cook, in some form or another, copies.

In the professional consulting and coaching industry, the essential thing the chef knows that the cooks don't is what it takes to change the culture. In my upcoming book, *Wicked Leadership*, I show how organizational culture is shaped by the mindsets, actions, and systems. Chefs know that none of the frameworks work all the time. Instead, they help others learn the underlying thinking and principles (Systems Thinking, Complexity Management, Lean Thinking, Psychology, Culture, etc.), then co-create a system that works best in their context. Chefs create more chefs. Is it in our DNA to be chefs? How many people do you know that want to be a chef?

In complex environments, we need to work to evolve our recipes as our ingredients change.

In the culinary world, there's nothing wrong with being a cook. Most people are cooks because, for most people, inventing recipes isn't a goal of theirs. Chefs are iterative and constantly evolving their recipes. When it comes to the reasoning "recipes" or frameworks we use to run our organizations, we may want to think twice about where we are on the cook-chef spectrum.

The Last 20 Years

Simply being a cook wasn't the intent of most of the Agile Manifesto signers. The Agile Manifesto was written in February 2001, almost 20 years ago, as of this writing. In the intervening 20 years, organizations of all sizes, from the largest enterprises to the nimblest of startups, have attempted to follow the values and principles of agile. Agile — spelled with the capital A — has informed frameworks, framed transformational business strategies, and, in some communities, elevated to the status of religious dogma. In other words, we've become stubborn in our beliefs that agile is the way. We're ever so sure that we're right, that they, whoever they are, are wrong. Have you noticed it?

We've become stubborn in our beliefs that agile is the way.

Agile Hasn't Fixed Our Problems.

As conceived by its creators, the entire premise of agile development was that its focus should be to deliver software and continually improve it to satisfy the customer's changing needs. Any focus that did not deliver valuable working software to the customer should be eliminated. Somehow along the way, agile got redefined by the consulting industry and what the majority of people (cooks) thought.

The truth is that most agile transformations and adoptions never delivered what they promised. Agile adoption failure stories are abundant. Just talk to anybody that's been part of one and hear what they have to say. A very large majority of all of these so-called "Agile" transformations and adoptions fail to generate the capability to adapt to changing market conditions. In this context, failure means that change initiatives that use agile approaches are reversed, canceled, or don't deliver the desired results.

Agile, as we know it today, hasn't fixed our ongoing problems. It never will. It was never meant to do that.

Regularly, thousands of coaches and consultants are asked if they could help bring agility and responsiveness to the organization. This leadership and organizational addiction to bringing in consultants and coaches to fix their problems have been around since the 1920s when James McKinsey and Marvin Bower injected the consulting industry with a professionalism that would shape it for the next 100 years. Bringing in coaches and consultants is like bringing in outside cooks and chefs to tell our kitchen staff what to do. The problem with this approach is that it only works until it doesn't.

We need to trust our people and not be dependent on consultants to tell us what to do. Do you know anyone that's tired of depending on external consultants? How about anyone that's tired of not being able to create things and watch them flourish?

New England Clam Chowder

To say that I love the New England Clam Chowder would be an understatement.

Made as it should be, is a dish to preach about, to chant praises and sing hymns and burn incense before. It has about a dozen ingredients that make it a signature dish here in the North Eastern United States. Similar to New England Clam Chowder, Scrum also has about a dozen ingredients (roles, events, artifacts, values) that make it a powerful framework for product development.

Now imagine that you are a cook who does everything by the book. But you don't have all of the ingredients for the New England Clam Chowder. Instead of clam juice, you add white wine, and instead of butter, you add coconut oil because you heard from another cook that those are good substitutes. Now we have something that looks like a New England Clam Chowder but doesn't taste like it. As a cook, you are not aware of how these new ingredients are reacting to other ingredients. I see this all the time with Scrum, where people pick and choose what they like from the Scrum framework. A few weeks ago, I did a workshop, and I heard somebody say that they're doing Daily Scrum every other day because the team has too many meetings. It might look like they are doing Scrum on the surface, but when we look at the outcomes and behaviors, it's not cutting the mustard. There is nothing wrong with using different ingredients if we know what we're doing. Just don't call it Scrum or New England Clam Chowder.

Why We Need Chefs

The new wave of agile is here, and guess what? It's not coming from the consultants. It's coming from the people and companies that have realized that to adapt to the changing market conditions, they need to create their own chefs and recipes.

In a fast-changing work environment, where "ingredients" are changing so rapidly, there is no single framework or recipe to address all our needs. If every team in our company is doing the same thing, we are not "big-A" agile or "little-a" agile. Instead, every team should be evolving its way of working. When we are truly agile we can change our direction very quickly — we can redirect individual teams and what they're building very quickly. What I've seen is that we need about 10–15% of the employees to be able to step in and occupy the full cook-chef spectrum. How do we learn to be more like the chefs of the world, who seem to carve their own way through life so effortlessly? We need both chefs and cooks. We need our chefs to help the organization create more chefs. Our internal chefs need to create something delicious from the ingredients available rather than blindly following popular recipes and frameworks.

If you start looking for it, you'll see the chef/cook thing happening everywhere.

Dare to be a chef!

This article was originally published at
https://baa.tco.ac/2_YI

About Miljan Bajic

Miljan is a very passionate Agile coach who truly enjoys challenges. As an Agile coach (CEC), trainer (CST), and mentor, he's supported many organizations in their Agile and Lean journeys. Additionally, he teaches Agile Leadership and Process Improvement at the University of Southern Maine and Organizational Change at the University of Southern New Hampshire.

Miljan is also a frequent speaker having participated in many international conferences and Agile events in Europe and United States.

Ways to Contact Miljan

miljanbajic.medium.com

miljanbajic@gmail.com

www.linkedin.com/in/miljanbajic

Best Agile Articles 2020

Beginning and Ending with Gratitude

By Ahmed Avais

I've experimented with gratitude for many years. It started with me wanting to believe its value. Now, there was this messy middle of my experiments where I would find myself frozen with inaction, insincere, and awkward. I had to build the courage to notice and appreciate others. To my surprise, it wasn't a noble deed for them, rather; it built an energy store for me. An energy store that I could use to great effect and impact.

Even in the face of extreme difficulty, there is much to be grateful for. Gratitude is a matter of choice, we can always focus on what and who we are grateful for. It helps us get grounded and build the energy to move forward in seemingly intractable situations. As a student and practitioner of complexity, I'm always looking for differences that make a difference. Practicing and expressing gratitude is one of those differences.

I was in China on an international assignment for 3 months and conducted several workshops for the coaches. I ran a gratitude workshop that I learned from Dr. Glenda Eoyang near the end of my assignment. I knew it would be impactful, what I did not know was that it would create ripple effects of positivity. One of my associates there recently reached out to me and was delightfully describing how she loves it and how she's facilitated in different settings including an extended leadership team meeting. She felt it spoke to her and that she's finding the same spark light up with others. (I intend to write up instructions for the workshop in a subsequent blog post)

None of what I have said so far should be earth shattering. It's common sense, it's simple. Yet, I find people surprised when I thank them for something they did or for the bright smile they brought to work today or that I'm just grateful to see them. It's simple, and yet, it isn't easy without deliberate practice.

One of my mentors helped me realize my own limitations when he asked me to thank five people every day until it becomes an unconscious habit. And man, was it hard. I would make excuses, say that it feels weird. And as hilarious as it would sound, he said I should be already 50% done before I even leave for work by thanking people in my household. He wouldn't let me get away with excuses.

And I'm grateful to my mentor for that. When I offer reinforcing feedback or thanks to others, I build my energy reserve. I feel more positive throughout the day.

I'll leave you with a meeting warm-up that I love doing. And I encourage you to try it with your team.

Invite people to answer a question in the category of gratitude. You may ask "what are you thankful for? Or you can change it up for limitless possibilities. The advice here is to pick only one question and model the behavior by going first. Here are two examples I've used in the past:

What fills you with gratitude right here, right now?

As you reflect on the past (day, week, month), who do you feel thankful to?

And now it's your turn!

About Ahmed Avais

Ahmed Avais is an Enterprise Agile Coach, focused on Business Agility.

He brings his natural and insatiable curiosity to invite people into his teaching and learning. He uses a variety of tools and techniques, all focused on helping teams and businesses more clearly see their system, allowing them to unleash their potential. Ahmed is passionate about mastery, growth, complexity, agile games, and social responsibility.

Ways to Contact Ahmed

agile-7.com
aavais@gmail.com
www.linkedin.com/in/ahmedavais
www.twitter.com/ahmedavais

Scrum and Diffusion of Responsibility

By Alex Kudinov

On March 13, 1946, 28-year-old *Kitty Genovese* was stabbed to death in Queens, NYC, across from the apartment building where she lived. Later that year The New York Times published an article claiming there were 38 witnesses to the murder (primarily residents of the nearby apartment building) yet no one provided help or called for it. Although the NYT later *admitted* that the story "exaggerated the number of witnesses and what they had perceived," the impact was still significant, as Kitty Genovese's death and the NYT article gave birth to what psychologists call the "bystander effect."

In 1968, researchers *Bibb Latané* and *John Darley* built on this idea when they described a series of experiments in which they studied how a subject's readiness to help in an emergency was affected by the presence of a bystander. As the number of bystanders increased (even if only as perceived by the subject), the willingness to help dropped precipitously. This behavioral pattern has become known as the "diffusion of responsibility." More recently, Daniel Kahneman, in *Thinking Fast and Slow*, picks up on this "helping experiment" and discusses it in the broader context of the diffusion of responsibility, which is where I first picked up on the concept.

By now you are probably thinking, "Well, ok, but what has it got to do with Scrum?"

I would argue – everything.

Let's look at some factors that cause diffusion of responsibility.

Diffusion of Responsibility and Team Size

Let's start with team size. According to the *Scrum Guide*,

Optimal Development Team size is small enough to remain nimble and large enough to complete significant work within a Sprint […] Large Development Teams generate too much complexity for an empirical process to be useful.

I would also argue that increasing team size beyond the recommended nine members encourages diffusion of responsibility. As the team grows in size, each team member's sense of responsibility is likely to drop, as they start thinking that other team members will take on the responsibility of completing a specific task. Greg Barron and Eldad Yechiam write about this in *Private e-mail requests and the diffusion of responsibility* where they observe that *"the probability of receiving a helpful response is an inverse function of the number of simultaneous addresses. […] there are more responses to e-mails addressed to a single recipient, that these responses are more helpful, and that they are lengthier."*

Another potential reason for the emergence of the diffusion of responsibility in larger groups is the perceived sense of anonymity. It's been proved time and again, that people's behavior can change dramatically if they feel that the flurry of activities of a bigger social group anonymizes the individual actions. Look at some anonymous comments on any social media platform and ask yourself, if these individuals had a reasonable belief that their comments could be attributed to them personally, would they publish that comment anyway? In many cases, the answer is a resounding no.

While the chance of the perceived anonymity as a cause for the diffusion of responsibility is remote and its effect might be quite insignificant on the Scrum Teams, I would argue that we still need to keep in mind such a possibility and watch out for the associated behaviors.

Diffusion of Responsibility and Task Focus

Development Teams are cross-functional, with all the skills as a team necessary to create a product Increment. Scrum Master serves the Development Team [by] coaching […] in self-organization and cross-functionality.

Albert Bandura, the father of Social Cognitive Theory, emphasizes that managers assigning specific tasks to employees gives rise to the diffusion of responsibility by shifting the focus from the organizational or social (team) goal to individual task-oriented activities. When individuals are assigned tasks, not only does *Daniel Pink's* autonomy go out of the window, but also purpose — another pillar of the personal motivation factor. Purpose suffers as people narrow-mindedly focus on their role as "task doers" as opposed to their role as team members in the larger organizational context.

Personal assignments tend to blur the organizational goals or product-driven outcomes in individual minds. "Hey, it's cool," we think, "my goal is to get this widget working; someone else will make sure the whole thing makes sense for the customer."

Individual assignments on Scrum Teams are done for a variety of reasons. One anti-pattern I see repeatedly is when managers become part of the Development Team and assign tasks to team members. Not only does it go against the grain of the self-organizing behavior; not only does it eliminates any autonomy the team members might enjoy and get motivated by, but also, while narrowing down the responsibility for an individual task to a specific team member, it defuses the more important responsibility for the outcome of a PBI and shifts it towards the manager, who chooses to assign the work.

In some cases, the assignment is done for the sake of keeping people busy. Such action misses the point of achieving the desired outcomes — which is usually not staff busyness. In other instances, the assignments are performed based on team member skills. It takes away from collaboration and promotion of skill development, learning, and propagation throughout the Scrum Team.

The less evil version of task assignment is when Development Team members themselves pre-assign each task in their Sprint Backlog during the Sprint Planning event. As opposed to just-in-time collaboration and agreement on the ways of tackling a specific PBI, pre-assignment leads to a narrower focus for the individual team members and a potential loss of the bigger picture and responsibility for the outcome.

If you didn't have enough reasons for managers, especially those with the historically strong command and control mentality, not to be on Scrum Teams, here you have another one – their actions will increase the diffusion of responsibility.

Diffusion of Responsibility and Expertise

The *Scrum Framework*, being a framework for creatively and collaboratively developing complex products, provides several rules that should eliminate this described behavior.

First, the framework does not want managers to be part of Scrum Teams. I hear the framework purists sharpening their knives already. However, since, *"Scrum recognizes no titles for Development Team members, regardless of the work being performed by the person,"* and since the title manager still bears a lot of loaded meaning and associated responsibilities, I believe that manager being a part of the team is a disaster waiting to happen.

In hierarchical structures, people tend to diffuse the responsibility to those who are above them in the hierarchy, possess more power, or bear more organizationally mandated responsibility. Violating the boundaries of the responsibility that match the team boundaries by bringing a manager into the team's fold will inevitably skew the perceived level of responsibility. The perception of responsibility will shift from being evenly dispersed across the team members towards those with more positional and organizational power on the team. In

Separating Leadership from Leaders: An Assessment of the Effect of Leader and Follower Roles in Organizations Virginia Vanderslice notices that "the association of level of expertise or role and the amount of work required can cause people to feel varying levels of responsibility and accountability for their contributions."

"But Alex," you say, "the Scrum Guide says nothing about banning managers from being a part of the team!" And you'll be right — the Scrum Guide speaks nothing about organizational structures. However, for potential managerial candidates to the Scrum Team, I would pose the following questions, "Are you ready to check your managerial powers and responsibilities at the door when joining a Scrum Team? Is your organization ready for that? What impact will this leveling have on your identity as a manager? On the team's identity? Is your team ready to accept you as an equal? Will the team environment become safer, more collaborative and creative with a manager on the team?" If you answer "no" to any of these questions (and I would be shocked if you don't), then I would suggest that managers joining Scrum Teams undermine the Scrum framework and jeopardize the desired outcomes.

Second, the cross-functionality of the Development Team is one of the main concepts of the Scrum Framework. From the framework perspective, cross-functionality results in a team's ability to perform work necessary for delivering a Done Increment with minimal or no outside dependencies. The concept of cross-functionality in its turn heavily relies on the T-shaped (or Pi-shaped, or comb-shaped) approach to developing team members skills. Sole reliance on individual expertise does not bring enough focus on the necessary skill development.

Third, local optimization for efficiency ("busyness"), rather than for outcomes have similar consequences to that of tailoring work to specific individual skills and expertise. While utilization optimization is a topic for another day, one of its impacts is that it transfers the responsibility and accountability for success from the team and organizational levels (i.e., responsibility for outcomes) to the individual and task level (i.e., outputs).

Conclusion

If you still think that diffusion of responsibility is not a big deal, think again. In its various forms and shapes, it can lead to such ugliness as moral disengagement, increased beyond-the-reason risk-taking behavior, social loafing, and groupthink.

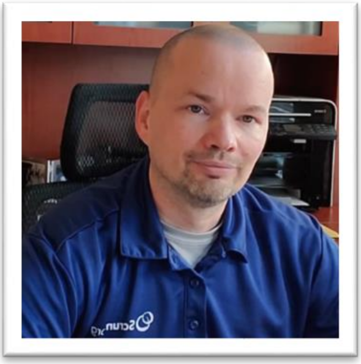
The Scrum Framework puts the responsibility for achieving the Sprint Goal and delivering a Done Increment squarely on the Development Team. Bloated team size, assigning work to the individuals, pushing work to the team, relying on unique positional, organizational, and skill expertise all lead to diffusion of responsibility and can be considered anti-patterns that contradict the letter and spirit of the framework.

Diffusion of responsibility is a real and quite nasty thing, and its consequences are potentially very damaging. That is how a meeting of excellent facilitators might end up with no facilitation or poor one. That is how teams might not reach their full potential, attributing the responsibility to individual team members or those outside of the team. That is how organizations might find themselves not achieving organizational goals as their employees are focusing on the parts of the whole, while the whole is not taken care of. That is how victims of an emergency might not get help where and when needed.

As discussed above, Scrum at the professional level has a few levers that either eliminate or significantly reduce the diffusion of responsibility effect. With that said, it is still prudent to keep the consequences of this effect in mind and be on a lookout for the diffusion of responsibility raising its ugly head.

This article was originally published at
https://baa.tco.ac/3Ej-

About Alex Kudinov



Alex is a technologist, Professional Coach, and Trainer. He delivers business competitive advantage to his customers by growing and nurturing great agile development organizations.

At Tandem Coaching Academy, where we keep agile coaching non-denominational, Alex is a Managing Director. We work with agilists of all walks, bringing agile and professional coaching closer together.

Alex holds the International Coach Federation Professional certified Coach (PCC) accreditation. He is also a Board Certified Coach, Certified Executive Coach, Professional Scrum Trainer with Scrum.org, Scrum Alliance Certified Enterprise and Team Coach (CEC and CTC) and Kanban University Accredited Kanban Trainer (AKT).

He holds an MBA degree from the University of Texas at Austin.

Ways to Contact Alex

tandemcoaching.academy

info@tandemcoaching.academy

www.linkedin.com/in/alexkudinov

twitter.com/AlexKudinovPST

Combining DevOps and Agile Transformations to Achieve Business Outcomes

By Andy Cleff

Is your organization well underway implementing DevOps? Or maybe you're just getting started with a DevOps initiative? Did you know you can maximize your chances of achieving desired business outcomes by combining DevOps with an Agile transformation?

You Keep Using Those Words…

When you hear "Business Outcomes," "DevOps," and/or Agile transformation" – what comes to mind?

We'll share our definitions. (If you disagree with our interpretations, contact us…we love a good exchange of ideas!)

Business Outcomes

It’s more than outputs.

These are the highest-level objectives of your organization. The big WHY. They are key inputs for your business and technology discussions around WHAT to work on. They are measurable outcomes – goal posts – that provide feedback on HOW your initiatives are doing.

DevOps

It’s more than continuous delivery.

DevOps is the practice of software development (Dev) engineers and of IT operations (Ops) working together during a product’s entire lifecycle, from design through development to production support, in order to shorten the total lead time (from concept to cash) and to provide predictable delivery of high-quality products.

Agile

It’s more than “Scrum.”

Agile is an iterative approach that focuses on collaboration, customer feedback, and small, rapid releases in order to satisfy the customer through early and continuous delivery of value. While the Agile movement originated in software development, it has been applied to much more: from medical devices to spacecraft, as well as engineering, marketing, and education.

Agile Transformation

It’s more than “Training, Titles, Ceremonies, and Tools.”

It doesn’t come in a box (or inside a cloud). It ain’t a silver bullet.

An Agile transformation is a rethinking and reworking of how your organization engages technology, people, and processes to achieve specific business outcomes. It is the relentless pursuit of continuous improvement.

What's the Pay Off?

Agile transformations and DevOps initiatives are complementary. Can you have one without the other? Sure. However, if you put the two together you have the opportunity to align the tech side of the house with the business side. This combination will enable your enterprise to gain faster feedback, reduce risks while also obtaining meaningful business outcomes.

We've identified nine common business outcomes, all of which are positively influenced by Agile+DevOps. (Reference: Harvard Business Review Analytic Services Survey, Sept 2018).

In our experience leadership tends to "Want them all, equally. And NOW" Sorry. If everything is important, you know the saying, nothing is. Prioritization of the organization's highest-level objectives, to avoid whiplash and to create focus, should narrow the field down to no more than three.

You then can measure the impact of your transformation initiatives against the top outcomes, providing actionable qualitative and quantitative data. (For more on metrics, see *Metrics in Agile: How to Effectively Measure Your Transformation Journey*)

Employee Engagement	Employees are more satisfied in their work, willing to go the extra mile, passionate about the purpose of their jobs, and committed to the organization.
Customer Satisfaction	Customers are satisfied with the experience, benefits, and outcomes when using your product

	or service.
Quality	The product or service meets the expectations of the market for usability, reliability, etc.
Speed	The time it takes to deliver an idea into the market.
Predictability	Teams maintain a predictable cadence of delivery enabling the business to make informed business decisions.
Innovation	New ideas, creative thoughts, or novel imaginations provide better solutions to meet new requirements, unarticulated needs, or known market needs.
Market Responsiveness	The ability of the organization to pivot quickly to respond to ever-changing market demands.
Productivity	Increase the business value realized while maintaining or reducing costs.
Continuous Improvement	The ability of the organization to relentlessly pursue optimizations in all aspects of business functions.

(We've got a poster summary of these *business outcomes that you can download from here*.)

What to Expect…

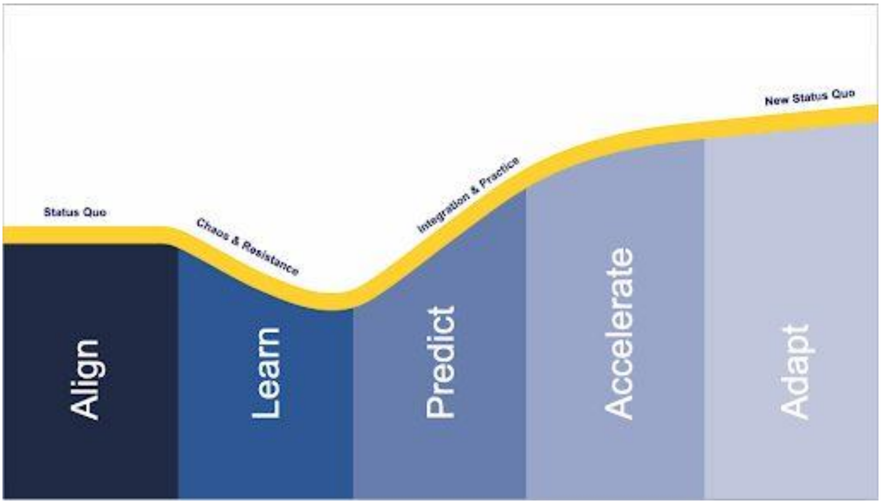
It will be challenging as there are many potential impediments that could derail the transformation without the right support.

From organizational silos to legacy technology. From the need to ensure security and compliance, to the lack of the right skills and even the right mindsets among employees.

The good news is that there's a path through. For over a decade, Agile Velocity has been helping enterprises chart and benchmark their progress during change initiatives. We've observed patterns of change, learning, and growth that move through five different stages, each building on the learnings from the previous stage. The Align stage is the first stage of the journey while Adapt requires a significant amount of agile maturity.

Align	Align the initiative with measurable business outcomes and define a clear transformation roadmap.
Learn	Establish foundational practices and a culture of learning by empowering teams to take ownership of their work and process.
Predict	Maintain a predictable cadence of delivery, enabling organizations to make informed business decisions.
Accelerate	Optimize the full value stream and shorten the time-to-market.
Adapt	Embrace organization-wide adaptability in order to quickly respond to market demands.

For software powered organizations (we could make the argument that 99% of all businesses fit this description), DevOps can significantly improve their ability to progress through the latter stages of Predict, Accelerate and Adapt.



Science has proven that introducing change into any system will result in a period of chaos until a new status quo is achieved. When adopting DevOps practices or undergoing an Agile transformation, organizations will experience a temporary decrease in performance before integrating new practices enables a new, more performant, status quo.

Many leaders don't acknowledge or plan for the "dip" that accompanies the learning stage – adding more change, producing more chaos, resulting in failed initiatives. Any organizational change must have leadership support along the entire journey or it will be short-lived or fail outright. With leadership support and by using the stages as a guide, time spent in "chaos" can be reduced.

Transformation Leadership – Where to Lean In

In their book *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations*, Nicole Forsgren, Jez Humble, and Gene Kim discuss the role of leadership and call it out as one of the more overlooked topics in transformation.

Like the authors, we believe that "Leadership really does have a powerful impact on results. . . . A good leader affects a team's ability to deliver … how the team manages its work and develops products. All of these have a measurable impact on an organization's profitability, productivity, and market share. These also have an impact on customer satisfaction, efficiency, and the ability to achieve organizational goals."

Some of the key transformation elements which require leadership involvement include:

Sense of Urgency

- Identifying and communicating the compelling reason(s) why the organization should change
- Ensuring alignment around the compelling purpose has been achieved at all levels

Roll Out Strategy

- Defining an initial transformation roadmap, one that takes into account organization structure demands, top risks, and incremental rollouts
- Aligning teams to value

Enabling Action

- Facilitating change to support the overall transformation
- Resolving organizational obstacles and impediments with urgency
- Continuing to communicate the change vision (rinse and repeat)

Cultural Shifts

All transformations require shifts in culture and mindset.

How big? The standard answer: "It depends!" Some situational variables are current organizational levels of

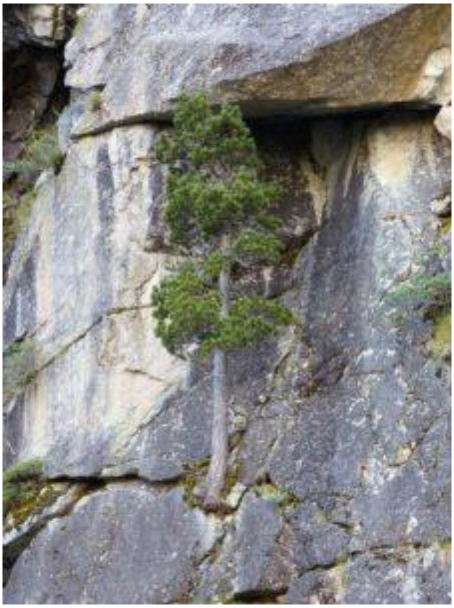
- Agile Architecture (vs monolithic, tightly coupled systems)
- Slack/Learning Time (vs emphasis on 100% "resource" utilization)
- Automation – Testing and Integration (vs over the wall or manual)
- Tolerance for Experimentation and Failure (vs perfection and/or blame seeking)
- Ability to Measure with Actionable Metrics (vs vanity metrics or zero data gathering)

For more see *Leadership Skills for the New Normal*.

This article was originally published at
https://baa.tco.ac/3DZt

How to Build and Sustain Organizational Resilience

By Andy Cleff



After reading the Gallup Group's article "*Remote Agile: Sustain Performance While Working Remotely*" a few things floated up in my mind. To keep those things from bouncing around too much, I thought I'd write them down and share them out.

In case you haven't read the article, here are Gallup's key points and recommendations for sustaining performance for remote and distributed teams at a glance:

1. The need for remote Agile happened overnight amid the pandemic
2. Build trust through relationships to combat anxiety in agile remote teams
3. Focus on four key actions to bolster a successful new way of working:
   - Maximize the use of virtual collaboration tools for remote teams
   - Replicate agile rituals while working remotely
   - Build trust in a remote work environment
   - Get closer to your customers

The Elephant in the Room

While there are some interesting (and debatable) points in the article, I think Gallup has missed a larger and more significant issue.

The issue is not about performance through tools and rituals. It is about building and sustaining organizational resilience–at individual, team, system and organizational layers–in the face of a VUCA* world, cranked up to eleven.

What is Organizational Resilience?

The dictionary definition–the capacity to adapt to stress or loss healthfully–provides a starting point. *Daryl Conner, in a blog post* expands things by providing a set of five “change muscles” that are needed to realize a capacity of organizational resilience:

- Positivity: seeing possibilities in even the most discouraging of situations
- Focus: knowing what’s important, and having a clear sense of priorities
- Flexibility: in the midst of ambiguity, complexity, and chaos, the ability to generate a wide range of options and ideas
- Organization: building plans, creating systems and structures to work effectively, and use their energy efficiently
- Proactivity: ability to carry out experiments with less than complete information

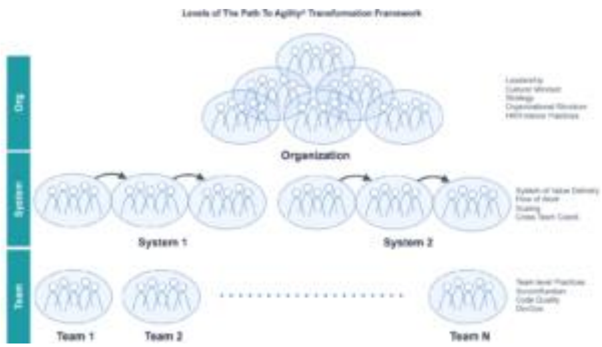
Mind the Wall

Based on what I’m sensing, hearing, and experiencing, many of us are at the equivalent of mile 18 or 20 of a marathon. Our *surge capacity* has been depleted. After running this far, we still have another 6 to go! To make things worse, there are no cowbells from the crowd to cheer us on and the finish line doesn't seem to be getting any closer. In fact, I'd swear someone keeps moving it out.

In these times, we are dealing constantly with the opposite of Conner's capacities: negativity, lack of focus, rigidity, disorganization, reactivity. We are facing "the wall" with prospects of burnout, as individuals, cascading to teams, and eventually hitting the entire organization.

Luckily, there are several approaches to addressing burnout in our topsy-turvy world:

- Individual plans (decentralized)
- Team and Cross-Team agreements (decentralized)
- Organizational policies and practices (centralized)



Start with Simple: Self Care

Individual plans are perhaps the simplest to get going. I'm experimenting with the following elements, both for my own resilience training plan as well during 1:1 coaching that I do:

- Routines: establishing a base of predictability in each day
- Grace: operating under the assumption that everyone, myself included, is doing the best they can given current circumstances
- Hobbies: Now that we "live at work" we need to make sure we're not all work and no play

- Connections: Planned serendipity with friends, family, colleagues, walks outside without screens/earbuds
- Boundaries/Constraints: Pomodori, WIP limits, one quarantini, not two or three, *the ability to say "No."*

A few nice things about the above "me, myself and I" components: they don't care about which branch of the org tree an individual sits on, and they can scale to team and system levels.

Team Working Agreements: We Care

When's the last time teams reviewed their working agreements to align with the new normal… especially taking into account the human elements?

One very powerful practice at a team level is to have check-ins. At the start of every interaction ask, "How are you doing?"– and when asked, provide an honest response.

Another: Express appreciation. What the person has done doesn't have to be a huge thing. In fact, it could be one of those tiny, everyday occurrences that may seem relatively insignificant yet made a difference to how the day, or even the past hour, went. For you, for the team, for your customers. *Read more about recognition, see: "Take 5"*

And a third (the power of three): Collective identification and *mitigation of the thieves of time*— becoming aware of and addressing the drag in the system.

Leadership Agility: Shaping the System

More than ever, leaders at every level need to remember that embracing Agile is not going to make all the problems go away; instead, it will shine a light on them. Those problems will still need to be dealt with.

Team Empowerment

Leadership agility unlocks the capability for many of those issues to be solved at the team level by ensuring teams have control, competence, and clarity. (For more on the topic of team empowerment, listen to our conversation with *David Marquet about Intent Based Leadership*)

Policies and Practices

Leaders also need to examine and re-shape organizational policies and practices to enable teams to better handle adversity with resilience and grit. These include HR policies around benefits, time off, flexible working hours, and more. For example, at Agile Velocity, we have expanded our vacation policy from one month to unlimited time off.

Developers of People

Agile leaders also need to shift in how they work with people, especially those who report to them. A key aspect is to create the conditions for success and guiding the development and growth of their staff.

Now more than ever effective 1-on-1's are needed. Leaders and their staff need to pause and think about what they've accomplished, to share how they perceive the world around themselves, and discuss what they have the potential to achieve. For more info, *see Ponderfy – "Mindfulness Hacks for Busy People"*

The Right Focus

Instead of a focus on sustaining performance, think about nurturing healthy environments that enable people to work through the tough problems that come with the complex (and chaotic) domains they travel daily.

If you focus on creating the conditions for a culture rich with organizational resilience, then higher levels of performance – (any way you define it: Employee Engagement, Customer Satisfaction, Quality, Speed, Predictability, Innovation, Market Responsiveness, Productivity, Continuous Improvement) will follow along for the ride.

*VUCA (volatile, unpredictable, complex, ambiguous)

This article was originally published at
https://baa.tco.ac/3ENY

About Andy Cleff

Andy is an experienced and pragmatic agile practitioner that takes teams beyond getting agile to embracing agile. His superpowers include holding multiple perspectives, enabling step changes that bridge current and future states, allowing space for both order and chaos simultaneously, celebrating success, and foremost, putting the people at the top of the priority list.

His chief weapons are well-asked questions, insightful retrospectives, and an ability to withstand awkward silences. And if all else fails, beer.

Ways to Contact Andy

www.andycleff.com
andy.cleff@agilevelocity.com
www.linkedin.com/in/andycleff
twitter.com/justsitthere

The Role of an Agile Manager

By Ben Kopel

When leading training classes or coaching teams and organizations, I often hear the question “what is the role of a manager on an Agile team?”

It’s a very good question and there are quite a few misconceptions. To start exploring this topic, let’s begin with what a manager typically does in a traditional environment. We’ll use Jim – who manages a team – as an example.

Jim The Traditional Manager

- *Output*: Jim’s main responsibility is the output of the team. Is the team delivering what is expected of them? They better be because Jim is accountable for the work getting done. This paradigm often leads to micromanaging, distrust and an ‘us versus them’ mentality.
- *Priorities*: Jim is also responsible for assigning work to the team. This often happens based on individual strengths, which does not support skill development and further perpetuates already existing silos of knowledge. It becomes easy for people to sit back and wait to be told what to do and keeps Jim really busy.
- *Performance*: Jim is in charge of who gets a raise, bonus or promotion. However, Jim is very busy monitoring output and doling out work, so it’s difficult to have an accurate understanding of how well each person is performing. Once again, this approach promotes the individual and not the team.
- *Capabilities*: Developing people’s skills is important but is often overlooked in a traditional system. When 1-on-1 discussions happen, Jim usually receives a status

report about the output and priorities, which feed into performance evaluations. Little time is spent coaching and mentoring.

- *System*: When it comes to improving the system in which teams operate, Jim does not have the time or authority to lead change. System level issues are often "escalated" to Jim's manager because that might be "above his pay grade." In addition, Jim does not have the knowledge of systems thinking and isn't sure how to connect the dots.

How Do Managers Become Managers?

Now don't get me wrong, I am not trying to imply that managers are incompetent or bad people. These situations and managerial approaches often stem from good intentions. Let's see if Jim's story sounds familiar to anyone…

Jim was great at his job – we'll call him a Doer. So great, as a matter of fact, that he was promoted to Lead Doer. After excelling in his role as Lead Doer, he was again promoted, this time to Senior Doer. And due to his knowledge, skills, and continued excellence in doing, he was promoted yet again. This time he was promoted to Manager… something he knows nothing about. In addition to not knowing how to manage people, he'll be expected to do it without any formal training. Jim is happy, but more than a little nervous…

If that sounds familiar, it's because that is the classic corporate career path of an aspiring professional. It's not really Jim's fault. He's just doing what' "he's supposed to do." It's a step I've seen many friends and colleagues take. And I've seen those people take it hesitantly because they knew they'd be giving up what they loved most about work. But it's a new challenge.

Unfortunately, it's a challenge in which most new managers are not set up for success. This is because many new managers don't receive training in management at all. If they do receive training, it is most likely in scientific management theory – also known as Taylorism – which has proven to be ineffective with knowledge workers.

Since Jim is now a manager, but hasn't been trained to be a manager, he will instinctually work in his comfort zone. When he was a doer, he was very focused on output and priorities. His manager would tell him what to work and follow up to see how it was going. Therefore, that's what Jim thinks he should do. Unfortunately, he is focused on the wrong issues. What needs to happen is that we flip the priorities on their head, and here's why.

Jim The Agile Manager

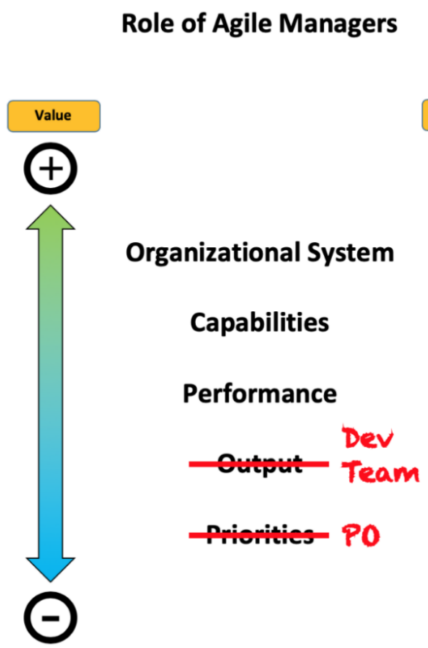
Let's imagine that Jim took some leadership training, got some coaching and mentoring and understands how an Agile manager should behave…

- *Output*: Jim is supporting an Agile team. A Scrum Team to be precise. While Jim used to focus heavily on the output, that is not necessary on a Scrum Team. Why? A Scrum Team has a self-organizing, cross-functional group of people called the Development Team. Their responsibility is to develop high quality products that delight customers. They were hired because they are good at their job, and the organization trusts them to deliver. Jim no longer needs to focus his time on managing the output and can redirect his energy elsewhere.
- *Priorities*: Jim used to let everyone know what they should be working on. They *relied* on Jim for that. However, on a Scrum Team there is a Product Owner whose responsibility is to prioritize the work. The Product Owner (and Development Team) collaborate with the stakeholders and users to understand what should be worked on next. Again, Jim can focus his

attention in other areas.

- *Performance*: While Jim will still be involved in assessing performance, many others will now be involved as well. On an Agile team, we'd like to see 360 feedback where the most important feedback comes from peers and those working closely with each other. This encourages accountability at the team level because teammates know who is contributing and supporting one another.
- *Capabilities*: Now that Jim is not focused on the output and priorities, he has more time to get to know what people are struggling with and how he can help them. Perhaps he coaches and mentors them. Perhaps he makes the case to increase the training budget so people can stay on the cutting edge of new technologies. Perhaps he finds some funding to support a community of practice. Jim is helping develop people and the team so that they can be more effective.
- *System*: Jim is now able to take a step back from the day-to-day details of the work being done by the Scrum Team and look at the whole system and how it flows from concept to cash. Jim learns about systems thinking and organizational design. He collaborates with other managers and executives to break down silos. As a team, they identify the system optimizing goal and work together to eliminate organizational impediments.

Jim is kicking ass and taking names. He has a new outlook on what a manager should do. He now understands that what he was once busy managing – the output and priorities – aren't the big challenges. The team can figure that out. His energy and



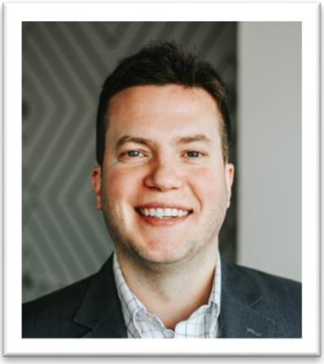
focus are better spent tackling the messy issues of politics and structures and policies that get in the way of real change taking place. Where people used to say "that's just the way things work around here," Jim is breaking down barriers and helping to change the organization's culture.

Traditional Managers vs Agile Managers

As you can see, the role of an Agile manager flips upside down the role of a traditional manager. Because the team is empowered to make their own decisions and trusted to get their work done, managers can focus on difficult issues that will lead to lasting and meaningful organizational change. Taking on those challenges requires skill, perseverance and fortitude.

It’s not easy, but someone’s gotta do it. And that someone is Jim, the Agile Manager.

About Ben Kopel



Ben Kopel is the Director of Agile Coaching at Project Brilliant. He is a curious learner focused on enabling adaptive organizations through awareness, collaboration and continuous improvement.

He has been working with Agile teams for over ten years as a developer, Product Owner, Scrum Master, Agile Coach, trainer and manager.

Ben is a Certified Enterprise Coach, Certified Team Coach and Certified Agile Leadership educator with the Scrum Alliance, as well as an Agile Leadership Journey Guide.

Ben serves on the CEC Review Team and the Guides Advisory Team, for the Scrum Alliance.

Ways to Contact Ben

www.projectbrilliant.com
ben@projectbrilliant.com
www.linkedin.com/in/benkopel
twitter.com/bckopel

Agile Coaches Need More Than Coaching Skills

By Bob Galen

I earned my CEC (Certified Enterprise Coach) certification with the Scrum Alliance in 2012. I'd applied in 2011, so it took me about a year to go through the process at the time. If my memory is right, I believe I was about the 43rd CEC at the time.

I did a search of CEC's and CTC's on May 26th and it showed 118 and 156 worldwide respectively. That's ~300 coaches. The growth of CEC's over an ~8-year period was ~10/year.

I had been serving in leadership and agile coaching roles for quite a few years prior to that. I'd also been part of a couple of fairly successful agile transformations at the organizational level. So, while I was confident in my coaching capabilities, I was excited to go through the process as a way of validating my journey, experiences, and skills.

Back then, the focus was on the breadth of your stances. For example—

- Coaching
- Mentoring
- Teaching
- Modeling
- Lean Skills
- Leadership Skills
- Consulting
- Partnering
- Facilitating
- Communication
- Business Skills

Were all areas where you had to have skills, experience, and acumen as a coach. And you needed to demonstrate a nice balance across that spectrum of capabilities. For example, you couldn't simply be a strong teacher or facilitator and weak in your consultative or coaching chops.

Instead, you had to be relatively balanced across all of them.

Of course, everyone has strength areas, but balance and breadth were the keys.

You can see an example model for what I'm talking about in this post about The Big Wheel of Agile Coaching. While it has a strong emphasis on coaching skills/capabilities, it's a much more broadly balanced and nuanced view to exactly what agile coaching encompasses.

But the times seem to have changed…

But I've discovered a potentially alarming trend in the Scrum Alliance (CTC, CEC) coaching paths.

There seems to be a strong skew towards the (professional) coaching stance. In fact, I think that skew is a dangerous one. Not because it's not important. It is. But because it's not the only stance. It's only one of many and balance and breadth are still keys from my point of view.

What evidence do I have?

Coaching Candidates

I've been mentoring candidates for the CTC and CEC since I earned my CEC. It's something I've been doing pro bono as a means of "paying forward" everything the Scrum Alliance coaching community has given to me.

What I've discovered recently is that there is a strong emphasis on professional coaching chops. In fact, a recent CEC candidate was told that they lacked PCC equivalent coaching skills and were not a good candidate. Even though they had attended iCAgile coaching classes, had solid coaching skills, and were incredibly balanced across the skill areas I mentioned. It seemed the emphasis was nearly all towards their ability to professionally coach—balance be damned.

And, as you'll see later, the PCC level is a fairly mature certification within the International Coaching Federation (ICF) certification levels, so not an easy hurdle to clear.

Guidance on the Scrum Alliance Web Site

You can also see the skew in emphasis in the documents on the Scrum Alliance website that detail the requirements for the CTC and CEC certifications.

- *SA – CEC Readiness Checklist*
- *SA – CTC Readiness Checklist*
- *SA – Study Resources for CEC Candidates*
- *SA – Comparison of CTC and CEC differences*

The starkest example for me of the skew is reading the study resources for CEC candidates. If you review it, there are three primary sections focused on coaching.

The starkest example for me of the skew is reading the study resources for CEC candidates. If you review it, there are three primary sections focused on coaching.

- *Coaching Tools, Techniques, and Frameworks, with ~15 references*
- Agile Knowledge, with ~10 references
- *Professional and Agile Coaching Competencies, with ~10 references*
- *Coaching Mindset, with ~10 references*
- Consulting with 1 reference
- Mentoring with 2 references

Outside of agile and coaching skills there are literally three recommended references. Imagine that…three!

For example, I think Gerald Weinberg's and Peter Block's seminal works on consulting would be invaluable references for agile coaches adopting a consulting stance.

And my point is not to expand the references now with recommendations. Instead, it's to show that they are imbalanced.

The Path To Coaching

A great group of volunteers on the Scrum Alliance have put together a series of recordings entitled the Path to Coaching. It's a wonderful resource to help develop your coaching stance skills. I haven't watched or engaged with all of the content. But I watched a sampling of them and reviewed the descriptions of the materials.

ICF coaching competencies were mentioned in every lesson I reviewed. And the teaching/focus, again, seems entirely focused on professional coaching.

Beyond what's been recorded, I'd love to see some coverage of—

- Consulting with leaders;
- Coaching UP and not simply from a coaching stance, but from an advisory or mentoring stance;
- Cultural transformation and scaling at an organizational level;
- The coach's role in Product Organizations and in understanding the business;
- Connecting to organizational silos, for example, HR and Accounting;
- Something focused on cultural tools like StrengthsFinders, Leadership Agility, and The Leadership Circle;
- OpenSpace Agility;
- Invitation/self-selection models and how to coach an organization towards those approaches;
- Much more in the way of storytelling; and while I'm there, what about communication techniques and storytelling for coaches?

And those are just a quick list of concepts and skills that are missing and would broaden the impact and value of the resources. But again, The Path to Coaching is a GREAT resource for learning the craft of professional coaching. It's just lacking other dimensions.

Professional Coaching

Now I want to clarify something. When I reference professional coaching, I thought I'd share some information as to what I mean by that. The following are representative of pure coaching:

- International Coaching Federation (ACC, PCC, MCC) credentials
- Co-Active Coaching (individual/personal coaching)
- ORSC Coaching (group/system coaching)
- iCAgile Coaching (well-regarded boot camp for basic coaching skills)

And all are representative of this coaching mindset—
The client is naturally creative and resourceful and whole.
Therefore, they have the means to solve their own challenges.

While I understand the power and simplicity of this mindset, when you adopt it too strongly over the other stances, you'll often miss other opportunities to help your clients.

Here's the language in the Scrum Alliance Comparison of CTC and CEC differences document—

- *A CTC is expected to demonstrate professional coaching skills at the ICF-ACC level. ICF certification is not required*
- *A CEC is expected to demonstrate professional coaching skills at the ICF-PCC level. ICF certification is not required.*

And keep in mind that both the ACC and PCC are fairly mature certifications with robust coaching training and practice requirements. And again, they are *purely* coaching stance focused.

Scrum Alliance

I did speak to a few folks at the Scrum Alliance about my concerns and I shared a draft of this post with them. Essentially the response (paraphrased) was –

We agree with you that perhaps we've skewed a bit towards professional coaching. The reasons for that are two-fold. First, there was a trend in the incoming CTC and CEC candidates where they lacked any sort of professional coaching skills. And second, we believe it's an incredibly important stance for all coaches and it's a relatively easy one for us to check on in early phases of the certification path. They also acknowledged that they're still looking for better balance in the agile coach's skill set even though everything on the website materials seems to skew towards professional coaching.

I do want to emphasize that they graciously took my feedback and promised to consider it in future adjustments in the focus for the certifications.

Wrapping Up

Independent of my conversation with the Scrum Alliance, I still think there is a problem. And I believe it's skewed beyond the Scrum Alliance certifications to the agile community-at-large.

It's not clear to me that every Agile Coach needs to have a strong coaching stance in order to effectively help their clients (our agile community). I would say there are situations where I would take an agile coach with strong consultative and mentoring skills and very lightweight coaching skills. They could meet a client where they are and help to guide them forward.

And I think there are situations where an agile coach, whose only strength is their coaching stance, might be incredibly unhelpful to a client by subjecting them to death by a thousand powerful questions. In other words, I can see senior leaders and leadership teams (ejecting) agile coaches whose only tool is powerful questions, clean language, NLP, or a variety of questioning frameworks in *every situation.*

What is that old adage, if the only tool you have is a hammer, then everything looks like a nail.

That perfectly illustrates my concern. It's not that the coaching stance isn't important for agile coaches. It certainly is. In fact, that's exactly why I've pursued my ORSC training and am currently in an ORSC certification cohort. To sharpen my saw when it comes to my professional coaching capabilities.

But that being said, I feel it's approaching irresponsibility to emphasize *any* stance/tool so strongly over the others.

Successful agile coaches need to be much more nuanced and situational in the breadth, depth, and usage of their skills. And yes, sometimes we need to be skilled consultants and *still* be regarded as "Agile Coaches".

I sincerely hope that the Scrum Alliance reconsiders its balance when it comes to the coaching certifications and immediately begins to swing the pendulum back towards the middle. Otherwise, we're certifying and releasing a large group of "Agile Coaches" on the business world who potentially have a single (albeit strong) skill in their toolbelt.

Nails beware.
Stay agile my friends,
Bob.

This article was originally published at
https://baa.tco.ac/2_LM

About Bob Galen

Bob Galen is an Agile Methodologist, Practitioner & Coach based in Cary, NC. In this role, he helps guide companies and teams in their pragmatic adoption and organizational shift towards Scrum and other Agile methods. He is currently a Principal Agile Coach at RGCG.

Bob regularly speaks at international conferences and professional groups on a broad range of topics related to agile software development. He is the author of the books: Agile Reflections, Scrum Product Ownership, and Three Pillars of Agile Quality and Testing.

Ways to Contact Bob

rgalen.com

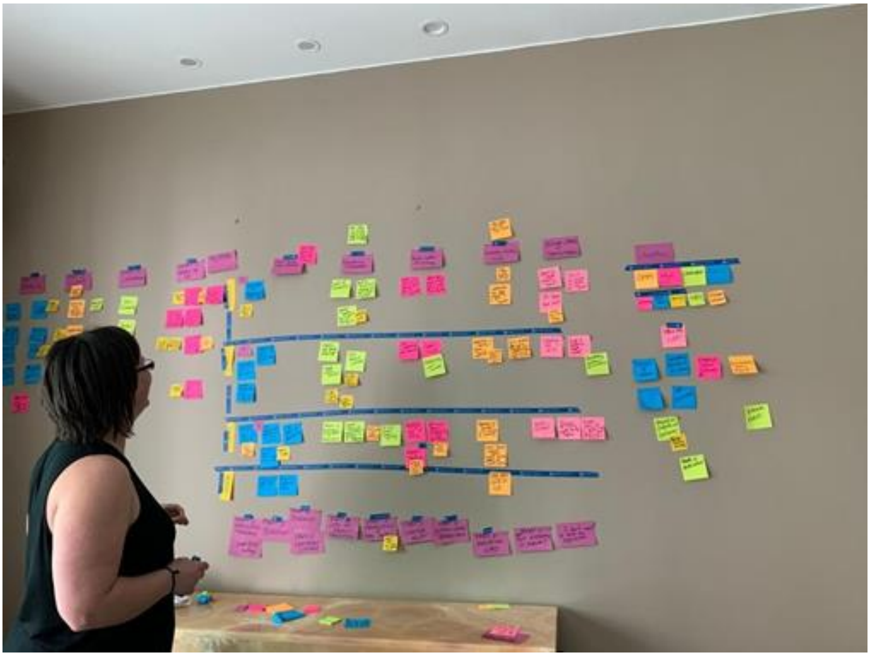
bob@rgalen.com

www.linkedin.com/in/bobgalen/

twitter.com/bobgalen

Escape the Productivity Trap

By Brandi Olson



"We had record breakthroughs in operational efficiency last year, but our client complaints went up." Margaret, director of client services at a workforce training center, went on to tell me that the chair of the board was thrilled with the cost savings while still being able to serve more clients. *"I think we are missing the point,"* she sighed.

Margaret's not wrong. There is an invisible and powerful cultural message that more with less is always better and that productivity is a goal unto itself. This is the Productivity Trap. The Productivity Trap mindset was at work in Margaret's company and the costs were growing, but none of it was showing up on the quarterly balance sheet.

Edward Deming called these kinds of costs "heavy losses." They are hard to evaluate and measure, but devastating all the same to the good *work we are trying to do and the people at the center of it.*

As an organizational designer and agile coach, I teach this, consult on it, and even write about it. And I still get sucked into the Big Productivity trap (you know, that productivity industrial complex peddling products and advice that only benefit the companies selling productivity).

In February of 2020, I was at the beach, on a writing retreat—with a wall full of sticky notes—my total happy place.

I created a sprint week because I knew I would need focus to do the work. It was fantastic. I had momentum, I was on track to wrap up my book that spring. I was productive in all the right ways.

I came home to a global pandemic spreading in the US. Within two weeks, my community was in crisis, my kids were home from school, and I had to temporarily shutter our family business. At the same time, my clients needed me more than ever, and I was rapidly learning new skills as a virtual speaker and facilitator so I could keep my consulting practice up and running.

And I was still trying to get up and write for the first few hours every day.

I added a healthy dose of regret and guilt onto everything else because I wasn't making progress on my book. I *knew* that there were good reasons I wasn't making the progress I had hoped for. But still, I suspected that if I were a true high performer, I could figure it out; that if I cared enough it wouldn't be so hard.

The real problem? (One of the real problems. There are many. These are complicated times.)

I was disconnecting my measures of productivity from the outcomes I was after. Let me say that again for everyone in the back of the room.

Productivity without outcomes is a trap.

I had an abundance of priorities—all real, all important, all immediate. I had even done the hard work of rank prioritizing—book writing landed about #5 on my list, and I was still beating myself up for not being productive enough. In reality, I had taken my one whole human self and attempted to divide my time among the eleventy-eight things that needed my full attention. In my plan, book writing was one of five top priorities. On my calendar, it had about 20% of my time. In reality, the writing was getting approximately 0% of my brain space—except for the brain space where I was feeling guilty about not getting more done on my book.

I was expending ALL of my energy, time, and resources. *Big Productivity would like me to believe that if I am using all of my energy getting items on my task list done, I am highly productive.* But even though I was checking off my writing time every day, I wasn't moving forward.

The Big Productivity almost had me. The thinking goes, if I can't make progress on my book every day, I am not being productive enough. But my actual goal isn't to make progress every single day. My real goal—the real outcome I was after—is to meet my deadlines so that I can publish on time. Publishing the book should be my real measure of productivity.

Effective productivity has to be connected to measures of success, and measures of success are highly contextual. I have a friend who's goal is to write every single day. She is working on building a writing habit. I am not interested in a writing habit; I am interested in getting my book out into the world, and daily writing time isn't going to get me there—not in 2020. I need writing sprints.

So, I am on another writing retreat. A 3-day sprint this time. All alone, out in a Minnesota prairie. I'm measuring productivity not by how many hours I write, but by how many chapters I revise. It took me most of the first day just to unwind my mind, and I had to fight the Big Productivity message that I was wasteful and unproductive with my time. In reality? I've accomplished my goal before the sprint was even close to being over.

What do you know? With singular focus and a clear outcome—I've been amazingly productive. I've accomplished more in the last 2.5 days on my book than I have in the last 3 months. And nothing has fundamentally changed about my priorities. There is still a global pandemic, my kids are still out of school, my world is still a roller coaster of uncertainty. But I am practicing the principles I teach—outcome-driven productivity and focus.

This book is being written a sprint at a time. I don't measure productivity by hours of writing, but by micro-outcomes—in this case, chapters done. Because every chapter done gets me closer to my goal of publishing. And the real, most important measure of productivity will be based on the outcomes after the book is published.

The shift to outcome-driven productivity is simple, but it's not easy.

It requires a relentless commitment to real work that matters and a keen awareness of the specious ways Big Productivity infiltrates our mindset. The question that guides me forward every time is "How will we know it's time to celebrate?"

I asked this question as I was sorting through how to measure my writing progress until I landed at an answer that was worthy of celebrating. Would I celebrate merely because I was adding hours to my writing log? Nope. What about getting as much done as possible during my writing retreat? That sounds great, but it's too vague. How much is "as much done as possible?" Would I celebrate my completion of three chapters on my retreat? Yes, that would get me closer to my real goal! And I knew exactly when to open the bottle of wine. I can only measure my progress and productivity in the context of a clear outcome.

This question is equally as powerful for leaders and teams trying to figure out their own outcomes.

Recently, I worked with a retail executive who was moving forward a strategic goal to process vendor payments more quickly. There were four product teams working on this effort, and it wasn't going well. The initiative was behind schedule, the teams were overworked, and the sponsor was frustrated. The executive wanted me to help this product group be more effective. When I arrived on the scene, the stated goal was to process payments faster. I talked with members of every team, and they each had a different interpretation of the success criteria. FASTER was all anyone agreed on.

"How will the teams know when to celebrate meeting this goal of yours?" I asked the leader. That wasn't a question he was used to answering, but it was exactly the question he needed to answer first.

Up to this point, the entire goal was to move faster. FASTER is nice, but that wasn't giving anyone what they needed to plan and measure progress. One design and planning session later, the leader and teams agreed that the outcome they were after was to process 90% of vendor payments in less than 10 days with zero errors. When can this group celebrate that they met this mission-critical goal? When the product enabled 90% of vendor payments to be processed in 10 days with zero errors. Now everyone had what they needed to have to measure progress and productivity.

Real productivity and progress can only be measured in the context of a clear outcome.

About Brandi Olson

Brandi Olson believes the best way to deliver out of this world value is through teams who are deeply engaged, high performing, and actually happy.

She works at the intersection between learning, organizational agility, and human-centered design and has over 15 years of experience consulting with leaders across diverse sectors who share a commitment to people, learning fast, and doing good.

Brandi leads practical consulting at The Olson Group and is the founder of Real Work Done, a learning and development company for leaders and their teams.

Ways to Contact Brandi

realworkdone.com

brandi@helloolson.com

www.linkedin.com/in/brandiolson

twitter.com/olson_brandi

Acknowledging Client Responsibility

By Cherie Silas

ICF recently released and updated competency model that will go into effect for credential applications in early 2021. The new model starts with the foundation: *Demonstrates Ethical Practice* which I wrote about last time, and Embodies a *Coaching Mindset* which is the focus of the next few articles.

ICF defines this embodiment of a coaching mindset as developing and maintaining a mindset that is open, curious, flexible, and client-centered. The first indicator that the coach is embodying the coaching mindset is:

Acknowledges that clients are responsible for their own choices

What does it mean to acknowledge that clients are responsible for their own choices?

I don't believe that this acknowledgment is intended to be just what a coach says to a client. The intention is not a checkbox whereby the coach says, "You are responsible for your own choices," though that may often happen in practice. This indicator is about the coaches acknowledging to themselves and accepting that clients are responsible for their own choices. This acknowledgment becomes a powerful influencer of the way the coach thinks, believes, and approaches her interactions with clients.

In order to more clearly develop an understanding of this first point, ask yourself this question, "How will I interact with others if I truly believe that they are responsible for their own choices?" Here are a few things that rise to the surface for me.

If I truly believe that others are responsible for their own choices, I will:

- respect that my client's way of doing things may be different than my own and recognize that they are the best judge of what is right for their life and situation
- be transparent, straightforward, and honest in my observations
- support their definition of success and partner with them to achieve that end
- hold accountability as a mirror so they can examine their decisions, actions, and inactions and determine the next step
- see them as competent, able to make the right choices and not in need of my fixing, advising, correcting, suggesting, or influencing
- believe that they can make the best decision to change or not to change and will respect their choice

I will not:

- become frustrated when they aren't moving fast enough, changing consistently enough, or when they seem to be moving in reverse
- push my own agenda, thoughts, opinions, and desires on them; nor will I try to influence them to see things my way
- judge their actions, inactions, decisions, or words to be right or wrong; instead, I will be curious about what they can discover through them
- try to rescue them, coddle them, or protect them

In short, this first indicator of coaching mindset goes far beyond what you do in a coaching session and encourages you to examine who you are as a coach. This sets a high bar and draws a clear distinction from how coaches with *ICF credentials* are responsible for being with their clients. It also sets clear expectations for clients about what they can expect from an *ICF accredited coach*. My question to you is, how closely are you aligned to this competency as an agile coach? Something to think about for sure.

About Cherie Silas

Cherie Silas, MCC is a Managing Director of Tandem Coaching Partners and the Director of Education at Tandem Coaching Academy.

She is a Certified Enterprise Coach (CEC), Certified Team Coach (CTC), and Human Systems Coach (CHS).

Cherie also serves as a professional coaching instructor and mentor coach helping coaches improve their core coaching competencies and skills. She is a Certified Supervision Practitioner (CSP) for experienced coaches helping them look at the work they are doing with their clients to strengthen those client relationships for more effective coaching engagements.

Ways to Contact Cherie

tandemcoaching.academy

cherie@tcagile.com

www.linkedin.com/in/cheriesilas

twitter.com/CoachingTandem

Do We Really Need A Code Of Ethics In Agile Coaching

By Chris Murman

Ethics: moral principles that govern a person's behavior or the conducting of an activity.

— Oxford Dictionary

One of my favorite improvements to running conferences (when we used to attend those in person) was adding a code of conduct. It was clear the behavior of a few dictated the need for everyone to play by a different set of rules. Granted, it hasn't halted all of the behavior it was meant to prevent. It cannot be denied, though, that progress has been made.

Combine that with an expanded view of what it means to be inclusive to all, the amazing work Women in Agile has done, and providing a safe environment for all, I must say how proud I am to be a part of this little family.

I was reminded of that this week when I discovered something that's been going on this calendar year. Not sure how I missed it before, but it seems the fine folks at the Agile Alliance are taking another step forward towards addressing ethics. *Shane Hastie* and *Craig Smith* are chairing a project to define a code of *Agile Coaching Ethics*.

Launched January 1 of this year, the intended outcome is to address a "lack of consistency in the discipline of Agile coaching and the lack of any ethical framework or code of conduct for that profession." It's a Big Hairy Ass Goal to be sure, one that immediately got my mind spinning. What does it mean for us to work under a single set of conduct guidance?

In the last week, the two wanted to advance the conversation by *asking what the community felt should be addressed in the code*.

Quick disclaimer: my company is a corporate sponsor of the Agile Alliance, and I'm a proud member. I've also chaired tracks for the Agile 20xx program team several times. In no way do these thoughts represent any of those entities. Just wanted to share my thoughts since Shane and Craig asked for them. Also to stoke more conversation on the topic.

What does a code of conduct for my profession mean to me?

First Off, Do We Need One In The First Place?

I dont mean to imply the request is faulty on it's face. When the conference code of conduct was first rolled out, I asked the same question. Mostly I was unaware that some of the poor behavior being displayed at our events, and once the stories of some were told on social media I couldn't have been more in favor.

So what behaviors are we trying to prevent in the agile coaching cohort with a code of ethics?

This group consists of independent consultants, trainers, full-time employees, consulting companies of all shapes and sizes, and curious individuals trying to constantly come up with better ways. It strikes me that each group might have a different factor driving their work.

Could I be an internal change agent trying to be the foreign element in Satir's change model? Perhaps I'm trying to show early returns on value for my work so I can get a contract extension. Conveying material related to a framework to achieve certification. I might want to improve technical practices within an organization that is at odds with business goals.

Each of those outcomes on their face dont have to be at odds with each other, but sometimes they are. If I can't even reach alignment on outcomes within my own company at times, how am I to align with the entire community. It could be conveyed the moral compass needs to be aligned first.

Can it be done? Perhaps, just curious of how we get there.

Do We All Agree What Coaching Actually Is?

When I meet with clients interested in the topic or agilists somewhere on their journey, we discuss what it means to be an "agile coach." If you peruse the Agile Coaching Institute website, in all of the references to the profession there isn't a ton of definition per se. According to CIO.com the role *is defined as*, "training corporate teams on the *agile methodology* and overseeing the development of agile teams to ensure effective outcomes for the organization."

Is that really what my job is defined as?

Inside the big industrial complex of the world of IT, I could see that angle. Many of my friends working on their own might not see all of that as their role. Perhaps they do, or agree in part. The crew at the Business Agility Institute are looking at a more *expanded view of people bringing agility* as the, "impact of change, both technological and cultural, is greater and faster than ever before…companies that are agile, innovative, dynamic, and perfectly designed to thrive in today's unpredictable market."

I could view my *responsibilities* according to the International Coaching Federation as, "to discover, clarify, and align with what the client wants to achieve; encourage client self-discover; elicit client generated solutions and strategies and hold the client responsible and accountable."

You might be reading all of that and hear the same message. Others might argue there are tons of nuanced differences. If we are to align on our ethics, then we should align on our mission first.

Once We Are Aligned On Those Two Ideas, What Else Are We Missing?

Hastie and Smith toss out some topics to consider as part of the code (in addition to asking for others. Exploring what they mean to us could provide insight into our ethical compass. While all the topics merit review, these were where my mind centered.

Working With Multiple Relationships: E.g. Sponsor, Team, Team Member, Manager.

This relates to my comment earlier about our underlying motivations in the coaching agreement. If there's no way to ensure the coachee wants you in their life, because leadership often makes that choice for them, it points to a damaged management of relationships.

While it is important to manage those relationships well in a good working agreement, it would appear many roles require this. I wonder if this could be related to anyone in our community (coach and otherwise), and thus a more broad code.

Reflective Practice And Continuing Professional Development, Including Self-care And Sustainability.

I've *written about this before*, but anyone calling themselves a coach who hasn't put themselves in an intentional position of being coached has to look hard in the mirror. How can I expect myself to know what it feels like to sit on the other side of the table (or Zoom call) from me if I haven't been coached?

A lot of the work could include meditation, mindfulness exercises, journaling, life coaching, continuing education classes, and mentorship. Admittedly, I've not been the most coachable person at times in my journey, but I can proudly state I've done all of those exercises. Sometimes at the same time.

It also relates to another bullet in the list about acting within your ability. If we put in the work to know ourselves and stay in a mindset of self-awareness, we can continue taking steps along the journey and act appropriately.

BTW, if you're up for a great life coach. Contact my friend *Pradeepa*. Tell her Murman sent you!

Conflicts Of Interest.

We are inundated with these conflicts every day in our work. If the manager is funding my work, is he my client? Are the teams whose work is meant to be improved? Perhaps the proverbial "value" that I'm supposed to help deliver daily? Those and others aren't always aligned. Choosing which one to rank higher have left me flummoxed on many days.

I want this to feel easier than it is often, but every coaching situation are as unique as the possibility of outcomes. Navigating this conflict leads me to vary my approach. A one-on-one coaching relationship often appear out of nowhere organically, and have the most opportunity for growth. However, when you balance their wants with the needs of their leadership and organization can muddy the waters.

Perhaps if we hone in on making the conflicts of interest visible and navigating those waters with open transparency, we are less likely to lead people where it wasn't meant to go.

Agreeing On Boundaries And Acting Within Your Remit.

I have said this many times, but I was raised very well by two amazing parents. However, I entered adulthood with a poor understanding of what healthy boundaries were. The fault for that lie in many places, most of all me not really being forced to confront what a boundary was. Just live by whatever rules I felt like making up on the spot and let the chips fall where they may.

Boundaries are a huge part of any professional success I've been lucky to have. Do I give of myself to those around me professionally without expending so much energy that there's none left for my family? Perhaps I look out for the needs of those paying my consulting bills at the expense of those who need my help (but didn't ask for it). Choosing to deliver a deck because it's easier can really swipe at my credibility.

There's also the boundary of wanting to just get along with everyone else and not ruffle feathers. Being a huge people pleader has often gotten in the way of healthy professional boundaries. Some might say, "who cares if you piss off the client if it's for the right reason?"

Understanding my boundaries and the constant movement of those borders can really set me up for success. It should also accompany the courage to defend them at nearly all cost.

Responsibility To The Profession.

This can truly be a deal killer if we dont remember what the point of our profession is and who we are supposed to be serving. When I enter engagements where I'm *not allowed to help understand and co-create expectations of success with the client*, I feel truly irresponsible to how I feel we are supposed to work.

It might also help if we identify what we could agree on irresponsibility to the profession means. Personal or hidden agendas could perhaps be added to this list. Craving the spotlight could lead to what I often refer to as "lone-wolf coaching." Does every agilist in the same area of business need to operate uniformly or can we all find our own way regardless?

This is crucial to understand, and honestly I'm looking forward to the exploration.

Finally, What Happens To People Who Dont Sign On The Dotted Line?

Nothing would make me happier to add my name to a list of ethical statements of our profession. Standing arm in arm with our community in the name of bettering our profession could be just the thing we need in this *messed up year*. Just curious of the rebels who dont want to play ball.

Agilists are notorious rebels. Hell, the manifesto was almost written with middle fingers blazing to those who didn't want to find a better way of working. I often find myself stiffening my neck when I'm told I can't or shouldn't do something. It's just a thing to consider when we ask people to sign on to something that could affect our standing in the community.

Would love to hear your thoughts and comments. Perhaps we could create an unofficial cohort of those brave enough to share with the two chairs. This blog is in effect my hand raising to participate.

Who's coming with me?

This article was originally published at
https://baa.tco.ac/3EJW

About Chris Murman

Chris's first job out of college was as a weekend sports anchor at an NBC affiliate. If he had only known what was in store!

Interestingly enough, he still loves telling the stories of others every day. Each interaction is an opportunity to learn what made you unique, and understand where you came from. If we got to know each other more on a personal level, it would make the tough conversations easier to have. Today, he serves clients as a principal consultant for Contino and is on the board of directors for the Agile Uprising Coalition.

Come tell him your story!

Ways to Contact Chris

chrismurman.com
chrisdmurman@gmail.com
linkedin.com/in/chrismurman
twitter.com/chrismurman

Why Face-to-Face Communication Is Not Thinking

By Cliff Berg

The *Agile* community is very big on *face-to-face* communication, and *rightly so.* I believe that the Agile movement's preference for face-to-face is a *reaction* to the 1990s era trend toward *thick documents*, whose authors - often consultants - had left the building, leaving programmers with impenetrable specifications that often contained inconsistencies and were too vague to be useful. The Agile movement also emphasized that one cannot create a complete spec up front.

I should qualify that. One *can* create an up front spec, but only with *enormous* effort, including simulation, so that inconsistencies can be discovered before the spec is finalized. But that is a tangential issue. The issue I want to focus on is face-to-face communication.

One source of misunderstanding is around *what "communication" between people actually is*. It is *not* an *event*: it is a *process*. For *complex* issues, the most effective communication usually *takes place over time*, involving a variety of formats: face-to-face, written exchanges, and diagrams or sketches. Back and forth.

How the process starts is a matter of preference and thinking style. In my experience, extroverts like to start with a face-to-face conversation, and see how far they can take that. *Introverts,* on the other hand, prefer to *start on their own*, first thinking things through in isolation, so that they can *think deeply*. Then they often like to have a few written exchanges. This is so that they can explain their thoughts without being interrupted, because an *introvert's greatest fear* about meeting in person is that they will not get a chance to fully explain their thoughts.

After thinking deeply, and after some written exchanges, an *introvert then is ready* to meet in person, to iron out differences of opinion.

What happens during an in-person verbal exchange? Well, it depends.

If you are like most people, *verbal exchanges* operate at a *very shallow* level of one's mind. It is basically *pattern matching* followed by *pattern generation*. That's why it occurs so fast: people sparring back and forth.

All they are doing is sharing what they already know. There is little actual thinking going on. I am not saying that *no* thinking happens - just not that much.

The advance of the OpenAI text generation system is an example of this. It is a neural network based system. Its latest version *can construct entire articles*, containing witticisms and what appear to be inferences. According to an *article in Singularity Hub*,

[The OpenAI system] performed well when tasked with translation, answering questions, and doing reading comprehension-type exercises that required filling in the blanks where words had been removed. They also say the model was able to do "on-the-fly reasoning," and that it generated sample news articles 200 to 500 words long that were hard to tell apart from ones written by people.

Yet its creators explain very clearly that *it does not "think"* - it does not "understand" the text that it constructs. It is only performing pattern matching and pattern generation.

It is not "thinking" - as in the kind of deep thinking that people do.

And what the OpenAI system does is what most people do when they *converse*: they are matching patterns and generating responses. They are not *thinking* - at least not *deeply*. It is all *very shallow*.

The benefit of face-to-face communication is therefore not that someone will attain a new understanding. Rather, they discover differences in how they look at something compared to someone else. They can then make a mental note to think that through later - deeply. Yes, they might make a new connection - and that is important - but they cannot think it through then, on the spot - so they are not yet committed to the new idea. It is just a kind of mental sticky note to think through later.

Therefore, *the hope of attaining lasting shared understanding about a complex issue, among many people, in the course of a single face-to-face meeting is futile*.

When someone *thinks deeply*, they are *constructing and trying new models* in their mind, and then applying those models to "simulate" the real world, to see if the new model can explain something that they have observed - such as one of those mental sticky notes. If the new model can explain the observation, the person mentally reinforces the new model, and plans to test it in the real world.

Thus, the formation of new models, coupled with mental simulation (execution) of those models, is how someone thinks deeply. This *requires focus* because it takes *a-lot of working memory* to do this: distractions consume working memory. It is not something one can do "on their feet" with others present, waiting for your response: there is not enough time to do it, and the presence of others consumes attention and therefore working memory.

Also, a new model that has not been reinforced through experience is fragile, and so *disruptions can shatter it*, requiring someone to spend a great deal of time and effort to restore it. That's why *people don't even try to think deeply if they expect to be interrupted:* it is not worth the investment of energy.

In other words, *if you put people in a busy room, they are not even going to try to think deeply* - not gonna happen, at least not unless they find a way to tune out the distractions, such as by putting two big screens in front of their eyes to block out visual distraction, using headphones, and ignoring Slack and email.

Some people - I believe mostly introverts - *prefer to try to actually think when they converse.* Consider for example *this exchange between Joe Rogan and Elon Musk*. When Rogan asks a question, Musk does not immediately shoot back: he often pauses and thinks - sometimes for several seconds.

If *Elon Musk* were not famous, and he were in a group, and paused like that, people would take the conversation and *he would never get a word in*. He would be the silent guy listening and rarely speaking.

That is the experience of introverts in most face-to-face group settings. They think before they talk, and so they can't get a word in, so they stop trying.

I know that I have equated introverts and deep thinkers, and that is not fair. Many extroverts are deep thinkers too. But I suspect - I emphasize *suspect* - that deep thinkers *tend* to become introverts to some degree, although there are surely many super smart extroverts. It is not an issue of intelligence: it is an issue of personality: one's analytical versus social inclinations.

In any case, I am talking about the *context of face-to-face conversation*. In that setting, it is the extroverts who speak up. Those *extroverts who are deep thinkers will find a way to get their thoughts out there* - they will make people listen. Their eyes go wide and they raise their voice and laugh and wave their hands; they rise from their chair and slap people on the shoulder and make jabs at others in fun, and make themselves the center of attention. It is how they are.

But if a deep thinker is an introvert, they will sit quietly and say little, even if they have the answer that everyone is looking for. They are waiting to be asked for their opinion, and they need to explain it in its entirety without being interrupted - something that just is not going to happen in a group setting.

So let's *not assume* that *face-to-face communication is the answer* to all our collaboration problems. It is not. Real, effective collaboration occurs through multiple mechanisms - mechanisms that are inclusive with regard to personality type. And real, effective collaboration about complex issues happens over time - not as a single event.

This article was originally published at
https://baa.tco.ac/3FDD

About Cliff Berg

Cliff Berg is an author of "Agile 2: The Next Iteration of Agile" (Wiley). He is a DevOps instructor and practitioner, and was an early adopter of Agile in 2000 when he was CTO and co-founder of a 200 person IT services company.

Ways to Contact Cliff

agilegriffin.com
cliff.berg@agilegriffin.com
www.linkedin.com/in/cliffberg

3 Agile Transformation OKRs Leaders Can Use to Avoid Wrecking an Agile Transformation

By Erik Cottrell

The leaders and change agents we meet are eager to ensure their Agile transformation avoids hitting rocks or running aground. After all, nobody wants to experience a shipwreck. We have found that the ones who do enjoy a successful journey have navigated three especially challenging obstacles.

We aim to address those obstacles in this article by offering three Agile transformation OKRs (Objectives and Key Results) for you to consider. For those who are not familiar with OKRs, they are a helpful goal-setting tool particularly for initiatives or efforts like an Agile transformation. OKRs are used to define how to achieve objectives (the "O" part) through concrete, specific, and measurable actions, the Key Results (KRs). The key results are time-bound and measurable milestones under the objectives.

We provide example measures below but keep in mind that your key results can and should be modified for your situation.

If you are just starting out on your transformation, it may be helpful to have more general measures until teams become predictable, at which time very specific measures can be used.

The First Agile transformation OKR – "Leader Engagement"

In the *14th State of Agility report*, the second greatest challenge experienced when adopting and scaling Agile is "Not enough leadership participation." The first is "General Organizational Resistance to Change." Both of these challenges are dramatically improved with increased leadership engagement.

An OKR addressing leader engagement is a great opportunity. Do well here and the impact can offer massively outsized returns. The opposite is also true. It's such a high-reward, high-risk area it deserves special emphasis.

Example Objective: Leaders are ready, willing, and improving their ability to lead an Agile organization

As measured by:

- Business outcomes are clear and understood by teams and stakeholders
- The transformation's compelling purpose permeates throughout the organization
- An Agile Leadership Team is in place
- The Agile transformation roadmap is clear and visible
- Leaders are trained and supported in their new Agile leadership capabilities

Organizations that have leaders who are ready, willing, and improving their leadership in an Agile approach recognize significant business advantages. When companies achieve this objective, as measured by the above key results, there is a new sustained capability that contributes to a massively positive impact.

These leader key results are intentionally chosen because they quantitatively describe how the ready, willing, and improving leadership objective is met.

These specific key results are taken from the *Path to Agility®*(P2A). With P2A, we lead Agile transformations by focusing on business outcomes which, in turn, engage leaders. Leaders who are excited about improving business results are far more willing to actively support a transformation effort.

Tactics you might consider offering your leaders:

- Apply a credible outcomes-based Agile transformation framework
- Utilize high-quality, tailored training your leaders will value (training to equip them for leading an Agile organization).
- Offer ongoing agility-based executive coaching to help them to enhance and/or build their new leadership capabilities.
- Have, and keep having, leadership conversations about the business goals and outcomes.
- Connect the business goals to the transformation effort so there is clarity about how the teams are working better towards measurable business results.

The Second Agile transformation OKR – "Team Capabilities"

What companies need are long-lasting Agile teams. Agile capabilities enable teams to adapt to the widest range of disruption and keep functioning at a high level. That is a promise of agility.

Keeping an eye on capability growth while adopting these new Agile ways of working is key. Good questions about Agile team capabilities are:

- How are the teams learning and improving?

- How are they collaborating better?
- How transparent is their work?
- How do they hold each other accountable?

Establishing these capabilities can be helped a great deal by implementing the necessary Agile practices. Also bear in mind that a team can do “Agile things” while not learning or improving, collaborating better, being transparent, or holding each other accountable, and so forth. So focusing on doing Agile things may not get you the result you want. The objective here is to keep the capabilities foremost in scope for long-lasting results.

Based on our research, most companies get stuck and stay stuck in middling Agile at best. What we observe is that it’s easier to measure how many teams are “doing Agile things” like how many “ARTS have launched” or how many teams are “having regular standups” and so on. Until relatively recently, measuring practices has been easier than measuring Agile capabilities.

We created the Path to Agility for ourselves, in part, to address the challenge of keeping capabilities in view. We needed a transformation framework that could establish business outcomes, identify and measure necessary Agile capabilities, and integrate Agile practices. It’s not an easy task to keep all those things in mind during a transformation. However, they are all critical for success and P2A keeps us on track for growing team capabilities.

Example Objective: Teams are ready, willing, and improving agility by building new capabilities

As measured by:

- Self-organization and collaboration as teams take ownership of their work (Team ownership)

- Clearly defined goals and aligned expectations enabling autonomy and understanding how their work ties into the larger whole (Team Purpose)
- Teams improving continuously to deliver value more effectively (Value delivery)
- Learning loops from experimentation with inspection and adaptation (Learning culture)
- Team work made visible (Transparency)
- Stakeholder input and feedback is sought early and often (Visibility)
- Team members seek ways to help each other to complete high-value items (Quality)
- Stable team throughput measures (Predictability)
- All team members are trained in overall <insert your Agile methodology here>

Granted, this a long list of key results, and it would be best to spread them out over time. This list is a selection of the kinds of capabilities a team likely needs to measure as they build mastery over time.

The teams that are ready, willing, and improving agility by building new Agile capabilities become dependable and predictable partners in delivering customer value. That is the significant advantage of team agility.

The Third Agile transformation OKR – "IT and Business Partnership"

One of the most rewarding Agile transformation obstacles to overcome is seeing business and technology work better together. When these groups partner and build trust, it pays important dividends throughout the organization.

We rarely see a greater incentive for momentum than when a customer lights up because a team solved one of their most difficult problems. When a team knows their work contributes to improving the customer experience, they are powerfully motivated. It builds momentum to higher team performance that improves over time. It's especially impactful when a cross-functional team sees how everyone's diverse yet coordinated work made it happen.

More than ever, the most difficult customer experience problems span silos and departments. To have the ongoing capability of solving those problems, cross-functional partnership and trust becomes mission-critical. Calling this out as a specific objective makes good sense.

Example Objective: Sustained partnership, alignment, and trust between the IT & Business departments

As measured by:

- Business partners at all levels are trained to support Agile efforts
- Customer and stakeholder feedback loops are shortened
- All teams are cross-functional with business involvement
- A clear value statement (or hypothesis) exists and cross-functional teams understand how they contribute
- Internal partnership scores are improving (e.g., eNPS)
- Cross-functional team are regularly retrospecting their progress and implementing improvement plans
- Predictability is improving.
-

These measures can be made more quantifiable to meet your needs. Each one represents a way to ensure partnership and trust are built for the long-term. They are all dependent on Agile's empirical process and benefit from the Agile approach of prioritizing human interaction. These often won't start happening unless new ways of working are introduced and encouraged.

When partnership, alignment, and trust between the IT & Business departments is present and customers lives are being positively impacted, the benefits compound.

Tactics you might consider offering your cross-functional business and IT teams:

- Have people from business and IT go through Agile training together so that they can learn together and develop a common language.
- Intentionally celebrate and enjoy customer wins together as a team.
- Offer soft-skills training (like bringing up difficult topics, holding each other accountable, etc.).
- Build a culture of learning by making the effort to discover what teams are learning and how they've decided to get better (rather than trying to solve their problems).
- Encourage team members to have conversations that help increase their appreciation for one another's challenges and obstacles.
- Use helpful tools like Value Stream Maps, Personas, Journey Maps, etc. to help team members see how value flows to customers and where that flow can be improved.
- Again, we recommend having a credible outcomes-based Agile transformation framework to align to common goals.

While customers stand to gain the most when high performing cross-functional teams work together to solve their problems, the internal benefits are also massive. When business and IT effectively come together, it sends a signal that all departments can, too.

Set Your Course

The success of your Agile transformation journey can be helped dramatically by using these three OKRs:

1. Leaders are ready, willing, and improving their ability to lead an Agile organization.
2. Teams are ready, willing, and improving agility by building new capabilities.
3. Sustained partnership and trust between the IT & Business departments

How you choose to describe, measure, and apply these three key Agile transformation accelerators can be as unique as your organization. The benefits to your customers, your employees, and the long-term results when you have them will be profound.

We wish you great success along your path to agility. Learn more about our approach *here*.

About Eric 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.

Ways to Contact Erik

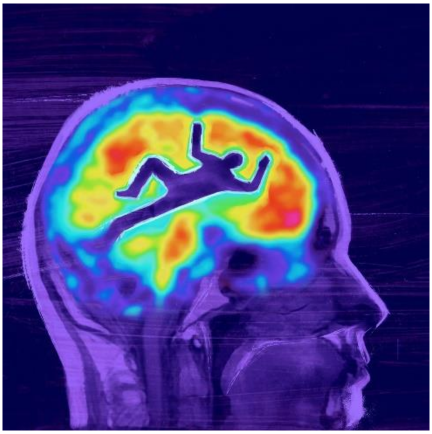
https://agilevelocity.com

erik.cottrell@agilevelocity.com

Why Neuroscience of change and Mental Agility are crucial skills if you are a Change Consultant, Agile Coach, Scrum Master, or HR

By Erich R. Buhler

When leaders ask me *"Will Kanban or the Scrum framework work here?"*, the most important thing I want to address is the leader's implicit premise that the framework will be getting better outcomes than in other competing companies. There is also something that needs to be considered here… Is this going to be sustainable? There is more to improving processes than just implementing new processes, and this is where mental agility can come in handy.



Organizational Health

When I say this, I mainly want to make sure they understand that whatever is done or suggested should not deteriorate organizational health.

From my perspective, Organizational health is Psychological safety—how people feel about talking freely and making decisions in the company—plus the creation of sustainable business value in perpetuity.

"Organizational Health: Psychological safety plus the creation of sustainable business value in perpetuity."

Here, the word perpetuity is crucial as it represents a behavior or practice that needs to be maintained over time. It doesn't mean that you have to use these words in your organization but, at least, the values and ideas behind it should be considered somehow.

For example, making people work long hours to meet a deadline or not having enough quality in something you build—if it is not done in a controlled and premeditated way—could make you get quick results but will not be sustainable. This obviously burns out people in the medium term.

These two concepts are key if you want to deal with continuous change and adaptation, and these two questions need to be addressed even long before thinking of implementing a new way of working.

In my experience, when a non-experienced Leader thinks of *Agile* or wanted to add new ways of working or doing things, it generally means in their mind that the same practices and processes across all circumstances, departments, and people will be established and normalized. This is a very interesting mindset because it can be used as an example of how they think of change and its culture works.

For example, if you are working with a large company that has been operating under the assumption that they have a great product and that customers will always buy it, then they might also assume that the framework will work well as a result of standardizing everything they do. This is an extension of the factory mindset toward the way of doing things.

If the leader is working instead with a small group of people in the organization, then their thinking may be different than that. And this is because she would be able to see that it's not about the framework and processes itself but how individuals increase their mental agility and embrace new values to solve the day-to-day challenges.

A must-have skill is to understand how people try to solve a challenge, and how they can temporarily embrace different points of views and values.

Reframing

This is what we call in Neuroscience of change, Reframing. Reframing is a powerful set of cognitive techniques that allow us to analyze a problem from different perspectives, guiding people to temporarily think as the person with the problem or from the perspectives of those who observe the problem, or even the individuals who don't believe there is a problem. This makes it possible to reach different conclusions and evaluate different assumptions.

Reframing is also important if you are going through a difficult situation as it helps regulate emotions and *diminish conflict*.

From the brain's perspective, it encourages new neural connections and results in innovative ways of thinking that help the company evolve. The crucial thing here is not about how the person sees the problem from another perspective, but how he or she can embrace temporary external values and principles that might be opposite to theirs.

The more "frames" a person can produce during a situation, the more flexible she or he will become. I believe that any company where their employees don't have adequate levels of mental agility would struggle to build sustainable innovation and would be less resilient.

Mental Agility

As you can see, competitive advantage is not just about frameworks but about initially helping individuals to get high levels of mental agility. That is why when I published my book, *Leading Exponential Change*, I created a model for Enterprise Coaches that places Mental Agility at the foundation of the company.

In my perspective, any organization that wants to have high levels of resilience and be able to produce innovation should have greater levels of mental agility.

Also, when the speed of change defies comprehension, conflict increases if the right practices, procedures, and techniques are not in place. So it is important to understand how mental Agility connects with the concepts of Exponential Change and Linear change.

Companies exposed to markets that are constantly changing require certain ingredients so that the desired behaviors become exponential—that is, for people to offer low resistance to change and be open to new ideas, with positive and proactive attitudes toward learning.

Employees should also be willing to modify how they think when they come across information that contradicts their beliefs or reasoning.

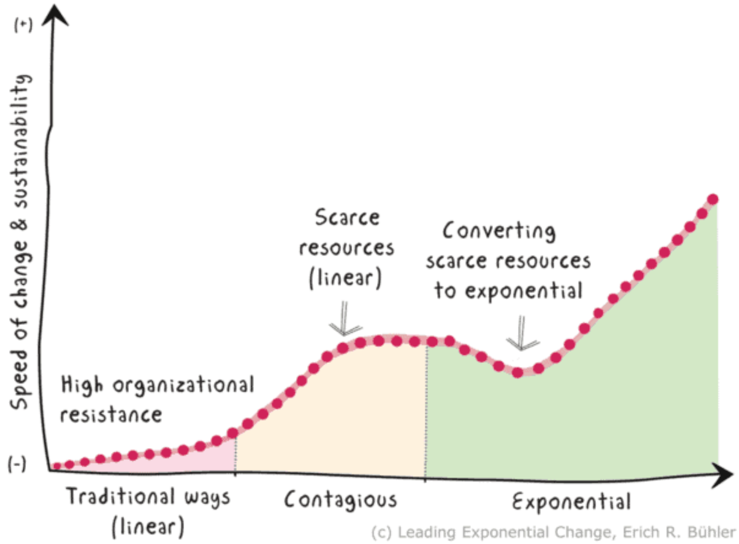


Figure 1: Ways to make a change and corresponding sustainability, (c) Leading Exponential Change, Erich R. Bühler.

Allow me to clarify the difference between exponential and contagious. In a company where change is contagious, nearby departments—uninvolved in the change—observe the change and adopt or copy some of their ideas or processes. If they find something interesting or useful, they will add it to their own toolbox as a new skill. But keep in mind that contagious mechanisms become very complex if you try using them to expand a practice, technique, or framework throughout the rest of the company.

In a business transformation where change is exponential, the result is different. Here, members of departments that surround those who are changing will feel that their own ways are outdated. They will fervently want to copy, understand, take over, and evolve the new processes. They feel it's no longer possible to maintain the status quo, that their world has changed, and that a door has opened leading to a better individual and group life within the organization.

To achieve this, the focus must be placed on increasing the mental agility of all the minds of the organization—and not just some departments in the company. People must be able to achieve a state of continuous well-being despite constant change.

Many firms begin by improving and making their software processes more adaptable (technical agility). This makes it possible to build a company more flexible and responsive to market disruptions. But while it might sound like a good initial strategy, it's not enough to increase the adaptability of the entire company (enterprise agility).

Enterprise Agility

To accelerate change, I developed a long time ago a change framework based on the Neuroscience of change that allows organizations to adapt faster and that is used by change consultants around the world. It is based on *five dimensions of Agility*:

1. Technical agility — Changing software as quickly, cheaply, and securely as possible.

2. Structural agility — Changing the organization's structures and procedures by running experiments while minimizing the impact on organizational health.

3. Outcomes agility — Delivering results even during turbulent times to respond to changing market conditions. This is related to leadership, budgeting, and how people and the company can change strategy and continue to produce great results with minimal impact on organizational health.

4. Social agility — Connecting well with other employees or with customers in rapidly changing environments, thereby achieving highly collective performance.

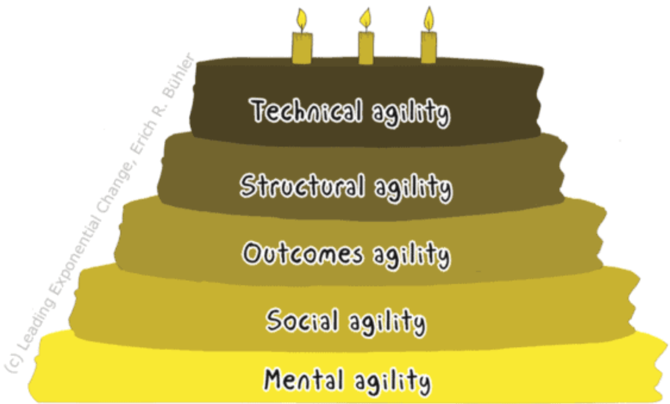


Figure 2: The five types of Agility in Enterprise Agility. (c) Leading Exponential Change, Erich R. Bühler

5. Mental agility — Reframing challenges to find new solutions even during stressful times. If the brain cannot accept emerging situations and realities or be able to reframe them using a high number of frames (mental agility), then employees will rarely adapt to new market conditions.

When the context changes, people must be able to find simpler and more innovative habits (or micro-habits) that allow them to better connect with each other, with customers, and with strategic partners (social agility). The company must be able to experiment and deliver results, even in times of high turbulence (outcomes agility).

The key here is what I call a Powerful change Plan. Any plan that wants to influence change and accelerate adoption over time should include these five different concepts. That means that the cake should be cut "vertically", and obviously you can't just "eat' the chocolate at the top when designing a change strategy.

This model is crucial to building an extremely resilient organization that is able to create innovation and adapt faster in perpetuity to disruptive market conditions.

I still remember, a bank with remarkably good agile teams (I.T.) I helped in New Zealand a long time ago was looking to expand the *agile mindset* and practices to the rest of the firm. They started building multi-functional groups all across the organization but somehow didn't work as expected.

Some employees previously worked on their own (financial and risk evaluation folks) or with individuals with similar mindsets and they didn't like the idea of working in new ways.

You need to consider that the human capacity to adapt to new ways of working depends largely on their personality, environment, and the type of profession they have been doing for years.

As you can imagine, placing the focus on the framework or in standard agile practices would not work well in that bank. The key to this challenge was to increase the mental agility of the employees and to make sure that they have good working agreements that promote healthy behaviors in perpetuity.

So when we think about influencing change in a company like this, you should consider how well people are at experiencing a high number of different frames (mental agility) when facing problems or new situations.

In my opinion, the more reactive you are without the capacity to reframe problems, the worse your chances of success will be.

Additionally, fear can reduce the number of frames the brain evaluates. For example, when a person feels that she'll lose prestige, power, or even her position in the company, that will activate an area of the brain called the amygdala. The amygdala is located in the center of your brain, which is where you store all your memories and information. It's involved in the formation of new neural connections between neurons. This process is called synaptic transmission. The amygdala controls our emotions, thoughts and it's aggressively triggered when you are under stress, fear, or situations where you might lose prestige or power.

It basically disconnects your high reasoning centers by flooding your head with certain hormones and prepares you for a fight or flight response. When this happens, your brain is on alert mode, and you are not able to produce the same number of "frames".

Powerful change plans based on five levels of agility and understanding Mental Agility and exponential change are crucial concepts if you want to influence change. Furthermore, all this happens even before you consider embracing a new framework or new ways of working.

I hope now when Leaders ask you "Will Kanban or the Scrum framework work here?", you have now a new toolset to help the organization be a better place

This article was originally published at
https://baa.tco.ac/3FDI

About Erich R. Buhler

Erich R. Bühler is the president of the Enterprise Agility University (EnterpriseAgility.University), a global organization that supports research and training to help professionals work

in post-agile companies.

He is passionate about organizational change and helped leading firms around the world to drive exponential.

Erich is the author of the best-selling book on Enterprise Agility, Leading Exponential Change 1st and 2nd editions, and creator of many of the widely-used frameworks for Enterprise Agility.

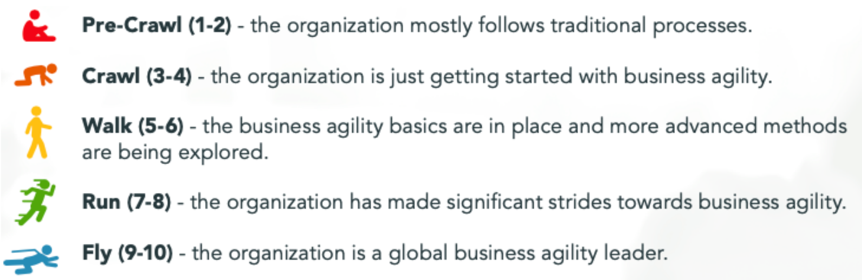
Ways to Contact Erich

enterpriseAgility.University
erichbuhler@innova1st.com
www.linkedin.com/in/erich-r-b%C3%BChler-09566067/
twitter.com/erichbuhler

Accelerating the Transition from 'Walking' to 'Running'

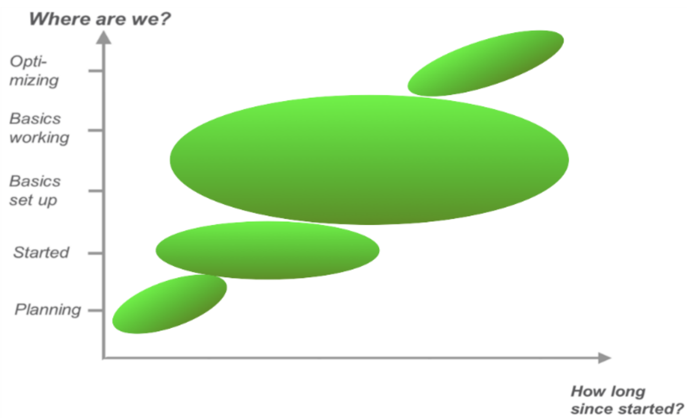
By Evelyn Tian

Back in 2012, I headed a group of internal agile coaches at Ericsson and supported different organizations on their agile transformation journey using our own internal coaching capabilities. We worked with 33 product organizations, each with, on average, a few hundred to several thousand people. Many management teams were curious to know what stage they were at in their agile journey, and they wanted to have an assessment that was more quantitative than qualitative.



While reading the Business Agility Report 2020, I was quite happy to see the increased level of business agility maturity measured by using the Pre-Crawl to Fly scale, which is reminiscent of a tool I have been using since 2012 to satisfy different management teams' needs and to trigger further improvement efforts. We aggregated data into a chart, to present the snapshot view of where some assessed organizations were at currently in order to encourage the internal reflection and increase leadership team awareness. It proved to be very powerful, and I have continued updating this chart over the years, from the angle of an internal agile coach and also after I founded my own coaching and consulting company.

First, let me start with a big disclaimer about this article. Over my close to three decades of work experience, I have spent most of my professional experience working in and with large corporations, so the data collected is primarily from large organizations (the entire company has over 10,000 employees, and/or the product organization has over 1,000 employees).



What prompts great reflections and discussions is the dimension of "How-long-since-started", tracking the number of years into the transformation journey, which is not limited to an "agile transformation". Most organizations are embracing agile values and principles by working with *Scrum*, Kanban or XP frameworks.

In the biggest green bubbles, there are organizations that might have arrived at optimizing but relapsed back to basics working, or from running to walking. You might have studied *GE's FastWorks projects*, and my experience has been similar. There are, unfortunately, relapses due to changes of leadership, organization restructure, and/or refocuses due to market situation and organization strategy.

Why Your Organization Is Stuck in the Basics of Business Agility

The wishful thinking that simply "implementing Scrum" can help bring about organizational agility.

Some organizations have been working with Scrum for years, and with a helicopter view, everything looks good. I keynoted at Agile Alliance XP 2019, titled "Agile, Mushroom and Tibet." The "Mushroom" covers this part: Without a clear strategy of why, what and how, all of a sudden, there are so many teams using whiteboards, practising Scrum events, sorting things in Jira, but many teams are just doing it.

Some organizations run Scrum-process. With the Project Management Office wearing the hats of agile coaches, the Scrum-process has been beautifully executed, but unfortunately lacks the key essences of continuous improvements and value-based thinking.

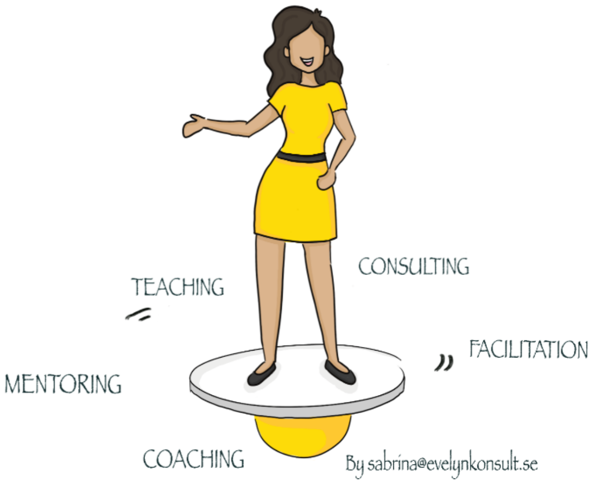
The pain of making true organizational change is so huge and bitter that some companies try to offer a magical remedy.

Many ideas sound fantastic, but with some deeper thought, they generate uncertainty, fear, and/or confusion. For instance, aligning the organization's structure toward business outcomes sounds truly great, though the implementation can be painful. How your status, responsibility, territory, ranks, and privileges which were built over the years becomes uncertain, at risk, or negatively impacted, and this is perceived as a threat and pulls you away from moving forward.

And you start to look for something less painful, with less bitterness. You hear about branded magic remedies, which lay out for you on a single sheet of paper what seems at first to be a simple formula for success. You take a quick glance at it, and suddenly your uncertainty, fear and/or confusion gets reduced and even naturalized. However, these magic remedies, like candies, bring in extra unwanted weights and consequences. It is just like gaining and losing weight - it is so easy to snack on sweets, but is so difficult to remove the unwanted weight afterward. So the original goal of improvement is revisioned into a new implementation of a framework--one that, despite its shiny promises, likely will not give you the results that you need.

Looking at the Glass Half-Full

I love to discover positive aspects of situations - imagine that we are positive and optimistic about the situation: What can be some improvement opportunities to accelerate the transition from “Walking” to “Running”? What can be done to help break the “basics working” plateau and move on to “optimizing”?



Now, if we have a pair of glasses with value-based lenses, full of curiosity and positivity, what can we see?

We see everybody focuses on value: value-based strategy, value-based thinking, and value-based agile coaching.

We see leadership teams and the entire organization being supported by agile coaches, and agile coaches balance their skills among teaching, mentoring, coaching, facilitating, and consulting.

We also see agile coaches actively assess the current context and decide what playgrounds they will work in. They may work with Scrum teams, *product owner*, technical practice, product-related (such as product strategy and business models), leadership team, organization structure, culture, processes, strategies, and/or continuous improvements. We see them actively grow their domain competency to best support the organization balancing the 5 skills as shown above.

Agile Coaching Prerequisites for Successful Agile

Transformation

- Agile coaches should know what competencies to grow and how they can assess the entire system and organization they are working with.
    - I have so far had the privilege to work with agile coaches from 68 countries through various programs. Before we started the program, many had their primary focuses on getting Scrum events up and running only.
- Organizations should build the awareness of right agile coaching competences and utilize the agile coaching capabilities to support the organization's value-based strategy.
    - Doing a search on agile coach job posting is like looking through a kaleidoscope: colourful but confusing. Different companies have their own tailored version of description based on their perception.
    - If you are a hiring manager, before you spend the time and efforts to hire an agile coach, reflect around what you are trying to achieve short term and long term. You want to have an agile coach to help you achieve your business related goals, and working with Scrum or being agile is one of them.

This is why I continuously advocate value-based strategy and value-based agile coaching. I have been speaking about it, and I will be keynoting at a couple of Agile Tour conferences later this year about value-based concepts.

To accelerate the transition from 'Walking" to "Running", to get out of the long plateau of "basics working", here are some tips.

Tips for Accelerating Your Agile Transformation

- If you are in a leadership role, move toward value-based strategy and utilize your agile coaching capabilities to gain greater value.
    - Constantly reflect on your strategy, which should be value-based, and make great use of the agile coaching capabilities you have in our organization. This will help achieve alignment in the organization and gain greater return of investment for the transformation efforts.
    - Value can be defined quite differently across different industries, and the exercise of discussing what value is at the leadership team level is a step forward. Then the visualization of the value stream can help reveal the system and serve as guidance for continuous improvements.
- If you are an agile coach, move towards value-based agile coaching.
    - Continue building the skills to allow the flexibility to use the right skills depending on the context
    - Continue growing knowledge and experience
    - Continue assessing the system you are coaching with value-based assessments and decision making, to bring the most value to the organization.
- If you are holding any other roles, actively apply value-based thinking.
    - You will help create a new experience for everybody around you, which will in term contribute greatly to the mindset change for

others around you.

With a move toward value-based strategy, you will start to experience greater business agility, shorten the years of "Walking," or in other words, break the plateau of "Walking" to finally arrive at "Running," and have even more fun transforming the world of work!

This article was originally published at
https://baa.tco.ac/3ElW

About Evelyn Tian

Evelyn Tian is a coach, trainer, consultant and keynote speaker.

She brings ~30 years of experience and a wealth of diversities - from product manager, PM, developer, architect, support to senior executive.

She coaches organizations from strategies, products to teams in various industries and cultures since 2009.

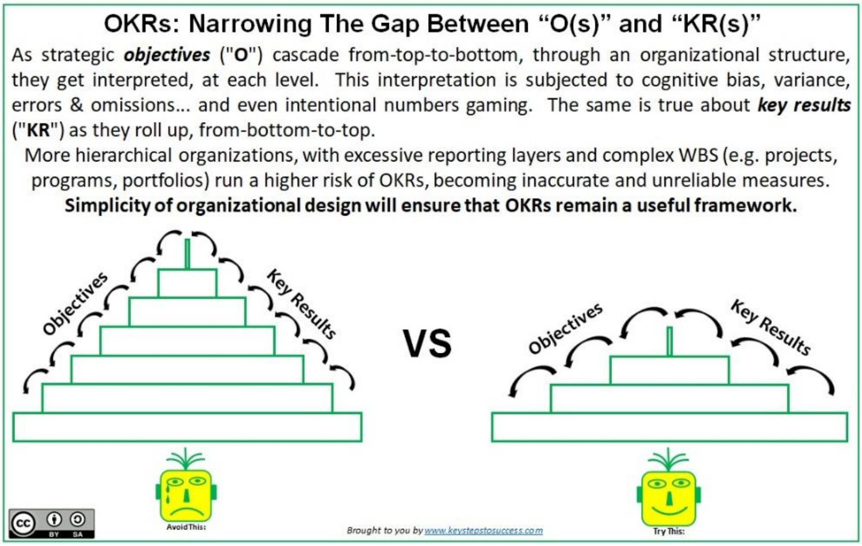
Evelyn is passionate about growing people and has supported practitioners from over 70 countries through her mentoring and training programs. Her value driven agile coaching cards are translated in more than 10 languages, supporting companies and agile coaches become more successful.

Ways to Contact Evelyn

www.evelynkonsult.se
evelyn@evelynkonsult.se
www.linkedin.com/in/evelyn-tian

OKR: Narrowing The Gap Between "O" And "KR"

By Gene Gendel



OKRs (Objectives & Key Results) – they could be your best friend or your worst enemy.

Albert Einstein once wrote on a blackboard: "*Not everything that counts can be counted, and not everything that can be counted counts. [source]*" This usually applies to things that people like to measure. But what does this have to do with OKRs?

OKR – is a way to measure *how* your initially set [strategic] objectives translate into results, at various organizational levels. With OKRs, there also come numbers, metrics, calculations, RAGs, etc.

What would OKRs look like for a traditional organization (complex structure, many reporting layers)? – this is not the scope of this writing. Frankly, there is a plenty written on this topic. Though, one thing is worth mentioning is that, as "Os" roll down from top (executive level) to bottom (individuals), and as "KRs" roll back up, through multiple organizational layers (managers, teams, departments), accuracy and relevance *decrease*, whereas variability/variance/inaccuracy and system gaming risks *increase*.

But what would OKRs look like for an organization that was trying to get away from a traditional structure and wanted to become more agile (adaptive)?

First, we need to define what it means, to be '*more agile*'? What is the most important factor that defines organizational agility? That is right!!! – *organizational design/structure* (culture, norms, values etc, come much later). When an organization becomes more agile, what we typically see, is reduction in a volume of processes, artifacts and single-specialist roles – people that are responsible for single, asynchronous tasks. This is sometimes referred to as *organizational 'de-scaling'* (flattening).

By the way, and just to correct frequent misunderstanding of the word means: agile – does not mean 'faster', 'cheaper', 'more efficient' etc. Of course, the ladder may come, as a secondary or even tertiary result of becoming more agile and responsive (e.g. a company, responds to market/clients needs than its competitors –> this leads to increased sales –> increased profits –> better economics) *but this should not be the main set goal of agile transformation efforts.*

Even at a large investment bank, with its traditional complex structure: multiple reporting levels, many applications (with application owners), many departments (with department managers), many sites (with sites coordinators) – in order to improve agility in e.g. product development, we need to de-scale organizational layers and bring closer together: real customers (or internal users) with development teams (GEMBA). Why?

Because, in a flatter organization, where customers and executive management set objectives, and development teams are expected to produce initial (low-level) key results, there would be fewer errors and omissions, due to a lower number of translation layers.

For example, a good business objective could be to increase an annual revenue by 5 million dollars. This could translate into a requirement of having an X-number of new customer-centric features that, if delivered by a certain date, would help generating more revenue for a company (as per forecasting by its market analysts). These X-number of features could now be entered into a product backlog (owned by a single product owner and multiple teams), refined and delivered incrementally, over time, sprint by sprint. The lowest level key-result, in this case, could be meeting teams' definition of done (DoD) and delivering a potentially shippable product increment (PSPI) at the end of every sprint. (*Please note that ALL teams responsible in product development would be sharing the same objective. By the same token, results would be measured for ALL teams – working together.)* The next level key-result, for example, could be delivering some large feature (or a percentage of an initially requested feature), by some arbitrary interim date.

The intention here would be to have as few translation layers as possible, while cascading up results from-low-to-high, to avoid errors and omissions that could be introduced during translation. Of course, this would only be possible if an organizational design *inside, around and beyond teams,* was simple (de-scaled).

Word of caution:

These days, 'agile' is one of the mostly *overloaded and misused* terms.

Agile has become one of the most popular areas, where transaction takes place and business is generated (recruiting firms, consultancies, tooling companies). The are many, commercially successful, complex frameworks and tooling solutions (with the word "agile" in them) that are marketed, as "unified & comprehensive solutions". This constitutes, what is sometimes referred to, as a '*Triple Taxation*' – and organizations become a victim of it. This is also a part of a bigger, industry wide *“agile” framework-tooling problem*.

Essentially, these approaches, redefine the “old world” with new terminology: projects / programs / portfolios become *agile* projects / programs / portfolios, PMO becomes *agile* PMO, BAs become *agile* BAs, and so on…

With electronic tools that are in close partnership with heavy, RUP-like frameworks (e.g. JIRA + SAFe), neither one of which are focused on improving organizational design, OKRs become just another *illusionary improvement*.

Companies should avoid falling into the trap described above. Aligning low-level OKRs (e.g. team project) to higher-level OKRs (e.g. portfolio), by merely, “mapping” numbers that are generated at single-team level (e.g. velocity or number of items delivered per sprint) through an “agile tool”, and processing them through multiple layers of projects, programs, portfolios, epics, themes, etc – this approach is prone to **unintentional errors and intentional system gaming**, and will ultimately lead undesirable outcomes, because fundamental problems, such as an organizational structure and its internal communication channels, are not being addressed. Such “cascading” of OKRs may look nicely on reports but because it represents a system that is flawed, it would not be valuable.

Conclusion:

OKRs could be a great communication tool and means of providing transparency and managing expectations. But OKRs are only as good as processes and dynamics they represent, with the ladder being fully dependent on an organizational (system) design.

In order for OKRs to be reliable and omissions-resistant, they need to be as simple as possible, and have very few translation layers between the highest-level "O" and lowest-level "KR". This is an organizational design question.

This article was originally published at
https://baa.tco.ac/3Efk

About Gene Gendel

Gene Gendel is an organizational design specialist, agile/lean coach and trainer, consultant and advisor to senior leadership with 20+ years of experience. Gene is dedicated to working with companies of various sizes and lines of business, trying to help them improve internal dynamics, organizational structure and becoming a better place to work.

Gene engages at all organizational levels: senior- and mid-level management, teams and individuals. In his work, Gene uses various methods, tools and techniques to amplify learning by others and to ensure that people gain autonomy after Gene "coaches himself out of the job".

Ways to Contact Gene

www.keystepstosuccess.com/about-me

gg_nyc@yahoo.com

Dear Executive... Inconvenient Truths about your Agile Transformation

By Heidi Araya

Most companies now either have jumped onto the Agile bandwagon, are considering it, or even in the middle of hiring consultants to help them "transform." However, *only 6% of companies think that they are "highly agile", according to a 2017 Deloitte study*. I'd like to convey a few thoughts to the executives and leaders considering an agile transformation, based on over 10 years of experience with many transformations and years sitting with executive leaders.

Dear Executive,

The vision and support for this transition is not something that you can outsource. It is not something "those people over there do." It'll change everything about the way the business operates, and it'll be challenging but rewarding at the same time. So read on for some thoughts…

It is a large effort for most enterprises. Undoubtedly, the larger the company, the more challenging the transition. Evolving from a business that operates like a machine (deliver pre-planned things and try to sell what we've built), to one that operates like a living organism (try experiments; inspect outcomes) is a huge shift in all ways -- management style, organizational structure, how the work is done, and what information is needed in order to succeed. It will not be easy, and it might take years, not months. Some companies start to transition several times, quit, and start again, sometimes years later. It is even more challenging for public companies, due to pressure of short-term stock market results. It is a continuous journey, not a destination. See the last point.

The current culture will likely be the biggest impediment. It is easy to force and push people to execute rote practices and tasks, but this is exactly the opposite of what will help the company succeed, and it will be a false sense of progress. Not everyone will be ready all at the same time! *Leverage those who are ready*, and use them as a launch pad to show successes. Use them as champions. Showcase the things they did to get where they are.

There is no one size fits-all. You cannot 'install' agility via a framework. The vision comes first for what business challenges it solves, or opportunities it can provide. Many companies start with a big bang approach, when it might be better to set up a few teams, give them the support they need, find and resolve the challenges that come up, and discover incremental ways to improve.

Provide the clarity of mission. You, the leader, must provide a vision in your organization for the problems agile solves for your business as well as the business outcomes you want, not practices that people follow in their day to day work. Why is making this change not only good, but an imperative for the business? What will happen if we don't change? (If you cannot answer the last question, start there).

Support. Leaders go first. Without executive buy in, deep understanding, and continuous support at all levels, the effort will fail. Everyone, from top levels down will need training and coaching to understand how it will impact their jobs and teach much-needed skills. Alignment on "this is the way we work" is required. Meet regularly with the transformation team or coaches and find out what they need in order to be successful. Send out a regular communication about the transformation efforts and accomplishments. This is about surviving, thriving, and remaining competitive.

Everyone must be agile. If leadership does not regularly reflect on what went well, and what did not, then why should teams? What does a continuous learning mindset mean for the leadership team? Set regular and productive retrospectives for all levels. Look for second order problems ("what are the conditions that led to this and how can we fix those conditions?" Rather than "why did this occur," and merely fix that first level problem). Don't make continuous learning empty words.

Use guiding principles and enabling constraints, not rigid rules to follow. Things like "quality over speed", "focus on finishing over starting something new," "sustainable pace," "understanding customer outcome before starting," etc. Rules cannot cover all scenarios.

A mindset shift will be challenging for many. Agile is about ongoing collaboration, openness to learning and failing, experimentation. People who think they know it all and are not willing to try something new, eventually won't fit into the organization. Make the desired outcomes clear for the leadership team and all managers, and you, the leader, should support them through the effort through training and coaching.

Your best people will leave if you don't work on making their lives *better.* People want *autonomy, mastery, and purpose at work*. A paycheck, snacks, and ping pong tables are not primary motivators. Oftentimes, without the desired autonomy and purpose, they leave for the competition. (1) Annual engagement surveys are not enough… Make sure that people have ways to surface their biggest impediments, listen to them, and act upon them. In some companies, these impediments to delivering work effectively and efficiently existed for years, causing disengagement and reduced productivity. Sadly, everyone knew about these, but usually no one ever gathered the organizational themes and measured the actual impact. (2) Push decisions to the lowest level possible. Invitations are engaging, as are decisions. *Invite your organization to uncover and solve its biggest challenges via OpenSpace*.

Transparency is key. Make the problems of the business visible to employees. Do customers find the features hard to use? Is quality an issue? Is revenue falling? Share this information with the organization to inspire them. Closed-door decisions and lack of key information are sure to help the transformation go backwards. Use storytelling format to make the information accessible and memorable. Sixty-three percent of stories are retained versus 5% of facts alone. Use a wiki. Work out loud in company online channels.

Agility has many facets, each with its own domain knowledge. All of these concepts will be new to your organization and people will require training in all these areas. The investment is significant, but the payoff will be even greater.

- *Complexity Thinking.* Instead of large, year-long projects planned in advance and executed by project teams, the need today is for small, cross-functional teams learning by inspecting and adapting their approach and what they work on, all based on feedback (Scrum framework falls into this category).
- *Leadership.* *A new kind of leadership is needed.* Command and

control will not help the company succeed in the future. The leaders who are there will have to learn to find a compelling vision, empower individuals with more decision authority at their level and in their areas of specialty, and consider group decision-making techniques to arrive at the best decisions. *Try open patterns of business agility.*

- *Teamwork.* Facilitating the creation of a highly-effective team is new for most managers of teams. There is a science to creating high-performing teams. Check out the *Google study,* the *Oxford Review special report,* and the *Flow System. Check out Open Patterns for tips on how to engage teams.*
- *Technical & Cultural Practices.* Things like TDD, BDD, CI/CD, unit testing, small vertical slices of features, pairing/mobbing, limiting work in progress, working towards business outcomes over single tasks, shifting left on security and quality, etc. will be new. These are often practices that seem counterintuitive but actually improve business outcomes. *John Cutler has a great draft book on the topic.*

It is a journey of continuous improvement. Agile transformation is not a project with a start and finish. It is a journey of continuous learning. Your organization will need to continue to evolve. This is the danger of setting fixed processes, without regularly inspecting what's working and what is not. The implementation of a framework does not grant one a magic agility wand. Try making one step to improve at a time, and reflect on it at a pre-set cadence (quarterly, for example).

It's a long and hard journey: Why it's still worth it for you

Leaving a Legacy. You can leave a legacy of an amazing place to work and enviable culture (even if you don't spend your entire career there!). It'll be a place of rich collaboration, true innovation, and interesting work at all levels. The people whose lives you have positively impacted may follow you, if you ever decide to leave.

Delivering products & services that delight. You will be working at a place where everyone wants to have your products, solutions, or services. Not only that, but you will be leading the team that makes it happen. You'll be at the forefront of an enviable organization.

Company survival. I've personally seen several companies die a slow death while struggling to understand market and customer needs, or prioritizing new features over quality or stable architecture. If you want your company to not only survive but outstrip its competitors, re-imagining the way the company works is required.

It is a moral imperative to change the way we treat people at work. Industrial age practices and management dehumanizes workers, treating them like cogs in a machine. *Worker engagement is low*. Most workers today have a completely different perspective and want meaning and purpose to their own existence, while contributing to their employer's purpose also.

About Heidi Araya

Heidi helps startups scale effectively and remain competitive, and established companies transition to "new ways of working" by building deliberately developmental environments... where we're not only improving the processes, but also allowing people to grow into new ways of thinking and being.

It's Heidi's mission to improve workplaces so that people are happy, engaged, and doing amazing work together. She is a co-founder of the Open Leadership Network and trains and speaks at events worldwide. Heidi also co-hosts a popular virtual meetup series for Agilists at *www.coachingagilejourneys.com*

Ways to Contact Heidi

heidi@brightlogicgroup.com
www.linkedin.com/in/heidiaraya
twitter.com/HeidiAraya

My Inner Game : Embarking a Personal Transformation Journey

By Jayaram Hegde

After working in Agile Transformation engagements closely over past few years one aspect which always fascinated me was the need of having 'Professional Coaching tool' to strengthen my journey.

Being an internal Agile Coach and working with Teams/People and Leadership closely, quite often I felt Professional Coaching would be a valuable addition to my 'Coaching' toolkit. Considering the future outlook was looking to take up Professional Coaching training program for last year and half.

After kicking off this journey few months back, can definitely sense a shift in me personally when I look back. Be it including the change in thought process and mindset, see myself embarking into personal transformation journey where I am trying to be a better version of 'my-self' each passing day.

What has really changed in/around me in this journey?

Self-Awareness (becoming self-aware)

Noticing my emotions,behaviors and triggers causing certain unfavorable responses has been a *BIG PLUS* in this journey.

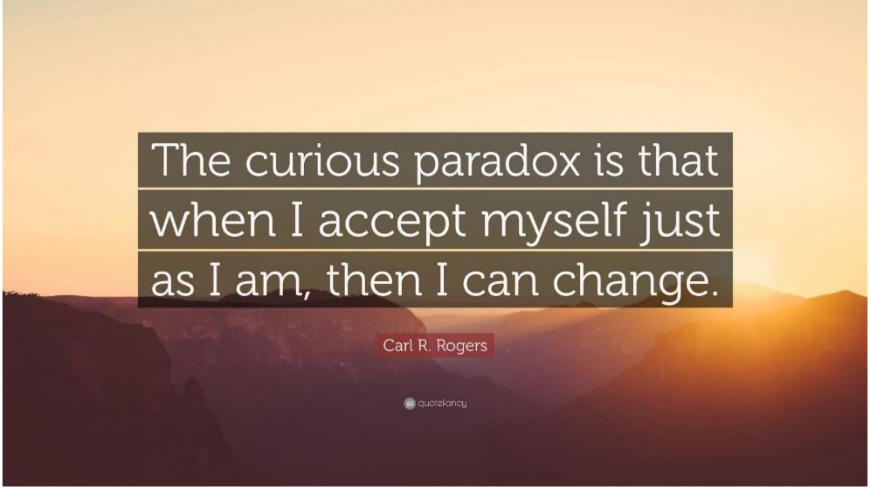
Being aware of my triggers & resulting emotions mean I can deliberate in how I can respond to others and to events — which makes it more likely that I can achieve something positive from the situation. By managing emotions and behavior, I can focus on finding solutions instead of dwelling on experiences, and feel more confident in tackling challenges. This doesn't mean I need to worry about suppressing emotions, for good !

Through all this I can first try to *learn from my emotions (input)* and *process the information internally (record/program)*, and use that information to choose *how to respond/react (output)* to situations.

Though working on managing the Triggers and being aware of the resulting emotions is a continuous journey, realized that staying positive, calm and appreciating things is nothing but *'sheer bliss'*.

When I look back, recognize the importance of maintaining calm, non-judgmental & positive frame of mind than reactive & judgmental way of situation management.

The following wonderful quote from Carl R. Rogers — father of Psychotherapy resonate very much at this moment.



Authenticity — True to 'yourself'

During this journey I realized the need of trusting people, because they know what is best for them.

"People are just as wonderful as sunsets if you let them be. When I look at a sunset, I don't find myself saying, "Soften the orange a bit on the right hand corner." I don't try to control a sunset. I watch with awe as it unfolds."
— Carl R. Rogers, A Way of Being

Recognized, the need to embrace the curiosity, asking more & talking less, being non-judgmental, trusting there abilities, holding yourself back, practicing authenticity and most importantly being aware of yourself (Self-awareness). Then only as a Coach, I can be a sounding board and/or a mirror for the client/coachee to invoke awareness and bring Self-reflection in their way forward path.

Every interaction with people has helped me to become better aware of my thought process, judgments and unconscious/conscious biases. Once I am aware of these elements, very next moment I could navigate myself and look at the person as is.

Focus on the person and not the problem aspect has helped immensely to stay with the person, connect with empathy, appreciate their learning/awareness and in turn helping me to listen better. This only has assisted me to look at myself deeper & deeper with each conversation and practice Active Listening, passion and love/respect to people.

Through this journey, I have come to realize that Coaching approach is here to stay with me. More than a skill or approach it is going to help me in self-exploration, self-discovery, self-realization and self-management.

More importantly I can be a better version of myself, for good!!

"we cannot change, we cannot move away from what we are, until we thoroughly accept what we are. Then change seems to come about almost unnoticed."

— *Carl R. Rogers,* On Becoming a Person

This article was originally published at
https://baa.tco.ac/3ExC

About Jayaram Hegde

Jayaram is an Agile Transformation Coach by Profession and agile enabler by passion. He is an ICF Certified Coach, seasoned agilist and a Transformation consultant who has driven agile transformation at scale within multiple organizations as a change agent, trainer, facilitator, and a coach.

Jayaram is an active community contributor, participant and speaker at public meetups, gatherings, conferences.

Ways to Contact Jayaram

jairamsrs@gmail.com

www.linkedin.com/in/jayaram-hegde-95b23921/

Dependency Management – the Good, the Bad, the Ugly

By Julee Everett

Does your team struggle to get items to Done? Do they experience a high amount of spill-over into the next cycle because they are waiting on another team or another person? Do items sit in a blocked state and age out while waiting on other teams or people to complete work?

Dependencies are an epidemic in software development. There could be many reasons why - perhaps your organization has adopted an Agile framework, but you're not yet structured to support sustainable teams. You may have a strong reliance on vendors or specialists when you start your Agile journey. And, some large-scale systems changes require some level of dependencies. The reality is, dependencies are not going to go away. As you scale your Agile efforts, the dependencies scale as well. The good news is, there are strategies you can use to move your teams from managing dependencies to mitigating dependencies.

What the Scrum Guide says:

"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." - K. Schwaber and J. Sutherland, Scrum Guide, 2013

That's typically not the reality for teams new to Scrum, or organizations new to Agile. While cross-functional teams are the cornerstone of Agile, it might take a very long time for your organization to evolve. If you want to fully realize the benefits of Scrum, don't just manage dependencies, ruthlessly mitigate them. Read on to learn how.

Thing we hear (or say):

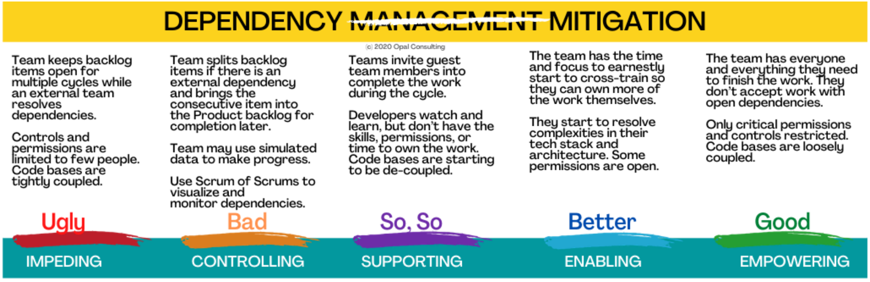
- Organizational Design: My team is mostly back-end; we rely on another team for the <…> (fill in with any component of software – front end enhancements, database layer, integration, API)
- Immature Agility: My team is using Scrum, but many of the other teams we work with are using more traditional approaches, like Waterfall
- Not Cross-Functional: My team uses a <Vendor / expert / lead > for a specialized skill, and they are in a completely different time zone
- Complicated Architecture: We work with so many different <systems, tools, technologies> that it's impossible to be truly cross-functional and loosely coupled
- What are your "reasons?"

The Good, the Bad, and the Ugly

As with all things Agile, there is no silver bullet, no one-size-fits-all solution. There are some things we can do, however, to mitigate dependencies. Some are quick fixes, and some require longer-term, more involved solutions. The solution to ruthlessly mitigating dependencies lies in empowering people and enabling flow.

Got ugly dependencies? Let's talk about getting to good.

In the short term, communication and proactive planning is key. For a longer-term, more empowering state, consider cross-training, opening up permissions, organizational design changes, even hiring and team forming strategies. It takes effort from everyone to see successes on a wide scale.



Short Term: Scrum of Scrums

A common approach to mitigating dependencies as you scale Agile throughout your organization is to use a scaling pattern called a Scrum of Scrums. In a Scrum of Scrums, delegates from each team meet to coordinate efforts and dependencies. A valuable practice is to visualize the dependencies and candidly prioritize according to the value and impact of the work. No pet projects or side jobs are allowed – all the dependencies must be made visible and evaluated against other demands. Another common practice is to ensure the decision makers are involved, so that resolution of impediments is fast and the group is focused on solutions.

If you are a team on the receiving end of dependencies, this is a good first step to relieve the need to attend multiple team meetings. But it is only one step on your journey to less dependencies - there is so much more you can do.

Long Term: Enable Flow, Empower People

Scrum Master or Tech Manager

Overall, be curious and ask powerful questions about each dependency. Don't accept the status quo - dependency mitigation is a complex problem without a clear solution. Open up a dialogue with the Scrum team, the leaders, even the Stakeholders to examine your work from various standpoints.

- Get work to "Done." If you aren't getting to Done, first, analyze what might be causing the problem: Are items too big? Are dependencies recognized during the execution of the work and not before? If items are just too big to be completed and fully tested within the time box, you may need to focus coaching and training efforts on vertically slicing small work items – with an emphasis on small. If the problem really is a dependency with another team or a specialist not on your team, discuss some solutions with your team, the Product Owner, and business leaders so they are committed to the behavior changes that are required to get items to a releasable state independent of other teams.
- Get the team involved. If the flow of work in your team has the hiccups, perhaps indicated by a high spill-over rate, lots of starts and stops on work in progress, or aging items, consider a retrospective on how to smooth things out. No-one knows better than the development team the art of the possible when it comes to using technology to solve problems. They are also the closest to the work, and they will have good insight to bottlenecks caused by permissions, controls, and expert knowledge only held by a few people.

Product Owner or Business Owner

Communication and preparation are key. So is saying 'No.'

One of the best things a Product Owner or Business Owner can do is to be proactive about planning dependencies, sequencing work, and negotiating scope. The leader's role in Agile to stay ahead of the team and to make difficult decisions every day is critical to the success of the team. *(Read my 5 tips to Emotional EQ as a Product Owner.*)

- Plan at least 2-3 cycles ahead so that developers can help identify dependencies early. That will allow you to work with any other teams and get on their backlog and ensure the value of the request is communicated. Consider using the various planning cycles in Agile to allow your team three 'touches' to decompose work, from vision and roadmap, to release planning, to refinement.
- Communicate the value, not priorities or deadlines. When teams have to balance multiple stakeholders with competing requests, we want to focus the conversations on the business value, not the deadline. Clarify for yourself and others: What is the business impact if this dependency is not resolved when we ask for it? What if it is not resolved at all? Is this dependency really critical to deliver something of value, or just preferable?
- Teams should not begin a work item with an external dependency unless the other team is at the Planning Meeting to make their commitment, or unless the dependency is resolved before the work starts. Be willing to sequence work that requires dependency from 3rd parties, so your team can build out their learning and skills to support more of the dependent work and lessen the reliance on others.

Development Team:

The Development Team is the first line of defense when it comes to dependency mitigation.

- T-shaped Skills: Start cross-training wherever and whenever possible, so that more of the team can do more of the work. Consider pairing in when a specialist is needed, by having that person join your team for a few weeks to share the knowledge and expertise.

- Accountability: Ensure that any refinement activities are hands-on, in the code, in the database. Take a personal interest in identifying where there is a reliance on other parties during refinement, instead of relying on others.
- Agile Engineering: Consider how technology can assist, such as automation, micro-services, or continual integration to release as frequently as possible. Refactor code so that your team can work independent of others. Ask for permissions or controls to be lessened instead of accepting aging constraints.

Agile Leaders:

Enable Flow, Empower People. The role of managers and leadership needs to evolve as well as the organizational structure to support the move to cross-functional, self-organizing teams. The criticality of this element can be overlooked while it seems all eyes are focused on development. But no-one else is as uniquely positioned to make system-wide changes that can free up resources and remove constraints.

- Enable Flow: Conduct lean analyses such as Value Stream Mapping to identify waste and delays. Your goal should be to evolve your organization into rational value streams and sustainable teams with a plan for less and less dependencies outside the value streams.
- Empower People: Create a Culture of Empowerment - are there single points of failure in your organization? Are there multiple levels of review? Is it a culture of centralized decision making or one of de-centralized, autonomy-driven decisions that creates bottlenecks and dependencies?

The difference between enablement and empowerment is that enablement focuses on the act of assisting, while empowerment is the granting of power to individuals or groups.

Empowering Product Owners and Business Owners to say "No" helps them limit work in progress, re-focus work on the highest, most valuable items, and eliminate the waste of coordinating complex, dependence-ridden efforts. Empowering Scrum Masters and Team Managers with more permissions and less reviews enables the flow of information and data. Empowering teams to say yes, and do more, helps the whole organization realize the full power and effectiveness of a unified and motivated workforce.

This article was originally published at
https://baa.tco.ac/3Daa

About Julee Everett

Julee Everett is a passionate and dedicated leader with over 15 years of strategic and transformation experience and an extensive background in management consulting, product, and Agile initiatives. She is a proven professional with strong relationship building and the ability to mentor others to ensure organizational success. Hands-on problem solver and an active leader in community and peer learning spaces, including Agile and PMP communities.

- 2021 Agile Alliance Agile Product Management Initiative.
- 2019 LiA 100 Women.
- Organizer of Tampa Bay Product Meetup and ProductCamp Gulf Coast.

Ways to Contact Julee

northstarproductlabs.com

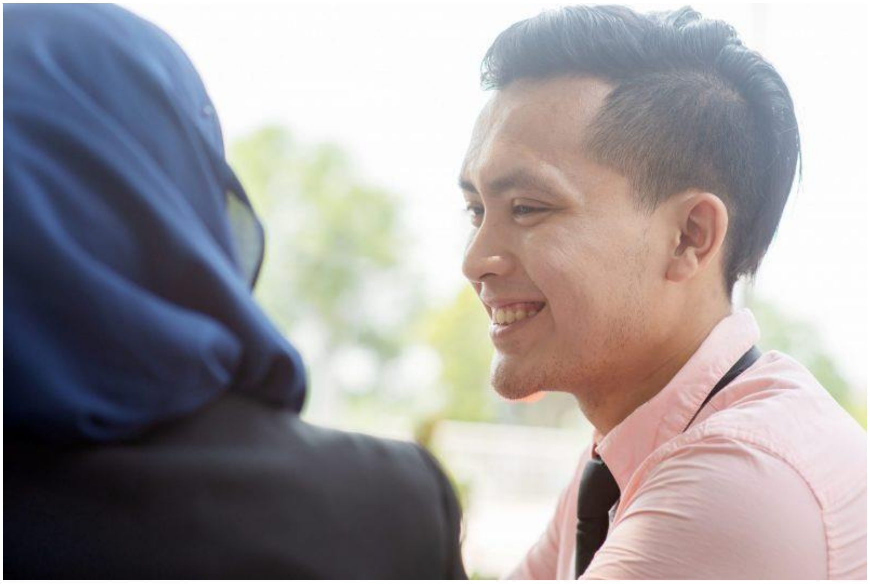
everett.julee@gmail.com

www.linkedin.com/in/juleeatnorthstarproductlabs/

twitter.com/Julee_NorthStar

The Transformational Factor in Coaching

By Kalina Terzieva



Why do we say that coaching can be transformational? As coaches, are we transforming the people we work with? What is actually the transformational factor in a coaching session? Asking the right questions ….

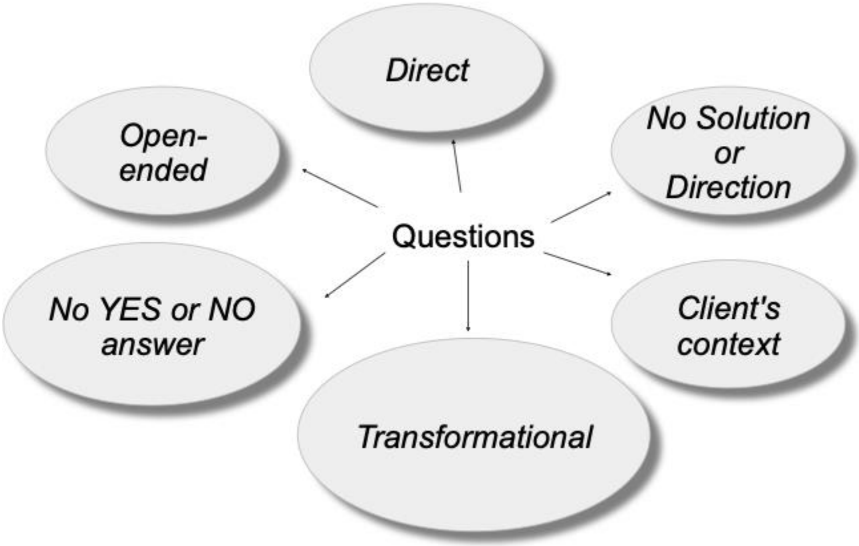
Coaching is a partnership between a client and a coach. Focused on the client's needs, the coach is asking powerful questions to make the coachee find the answers by himself.

TRUST is the key in this partnership. The coach trusts the client can find the answers they are looking for. The coachee trusts the coach and his ability to help and transform.

People will say–it is easy to be a coach. It is not …. The humans have the tendency to solution when there is a problem. My first occupation was software engineer. What a talented engineer does is finding solutions!

Now I am practicing my coaching skills and it makes me happy. When someone has a problem, the key is to ask questions that are:

- Direct
- Open-ended
- There is no YES or NO answer
- Starting with the magical letter "W" such as "Who" "Where" "What" "Why"
- Related to the client context
- Transformational



Direct questions are powerful because they do not let us make assumptions. Assumptions can be true or false, but as coaches we do not want to take that risk. Making assumptions means asking leading questions . This is not helping the coachee, though…..

Open-ended questions help the client think further. The goal of coaching is to help them find the right answer by themselves. Open-ended questions are helpful, not harmful.

Questions need always connect with what the coachee said before and the context of the conversation. Transformational questions help the client transform. Those are not transactional questions .

Transactional questions are like transactions of data between the client and the coach :

- They help the coach discover more information about the client
- They do not help the client move further
- The coach learns more about the client's situation and he will have the tendency to solution instead of asking powerful questions

When coaching, we can apply to following simple rules:

- Ask one question at a time
- Follow client's pace
- Add silences
- No leading questions
- We should connect every additional question to what the client says

Here are a few examples you SHOULD NOT follow :

- *"What would happen if you ask your manager to help you?"* This question has a solution and direction.
- *"Do you want to do that?"* This is a "YES" or "NO" question.

Here are a few examples you SHOULD follow:

1. *"Who can help you with this problem?"*

- Open ended
- Direct
- No YES or NO answer
- NO solution or direction
- Connected to client's context

2. *"What might help you with this problem?"*

- Open ended

- Direct
- No YES or NO answer
- NO solution or direction
- Connected to client's context

The transformational factor in coaching is the simplest thing in the world. Be an active listener and ask powerful questions.

About Kalina Terzieva

Kalina is an Enterprise Agile Coach and SAFe Trainer with 10+ years' experience shifting and transforming all levels of the organization to become more agile by influencing executives, championing methodologies, and guiding individuals and teams.

She delivers programs and training that influence entire portfolios on Agile, while supporting ongoing improvements.

Kalina is also mentoring and coaching scrum masters in their professional journey, helping them to reach their full potential.

Ways to Contact Kalina

kalina.trz@gmail.com

www.linkedin.com/in/kalinaterzieva

"How Do You Want to Spend 35-40% of Your Day at Work?" – Co-Creating a Team Culture

By Khurram Bhatti

On average a person spends 8.5 hours per day and according to various studies probably productive for three hours only.

How often do you reflect if you are in a growth environment

or contributing to an environment where you can grow. An environment where you are learning, experimenting new ideas, and having fun at the same time. Where you have psychological safety. One way I have experimented with creating psychological safety and designed team alliance is by talking about team values.

While coaching agile teams, by setting up expectations in a team has supported to create transparency in the team culture. While kick-starting a remote agile team a few years ago I wanted to present team values to the team however with a different flavour. That's when I asked the team *"How do you want to spend 35-40% of your day at work?"* and if that is an important conversation to have in the team setting.

Why talk about Team values?

Our values help us to understand what is important and what is not important in life. Values represent our individual and unique essence, it is our most fulfilling form of expression and relating to each other.

When we know more about our values it helps to find purpose. Values strengthen our ability to navigate through a difficult situation by choosing the right behaviours, actions, or interactions. Values help us to clear out clutter, make decisions, boost confidence, and increase a sense of self.

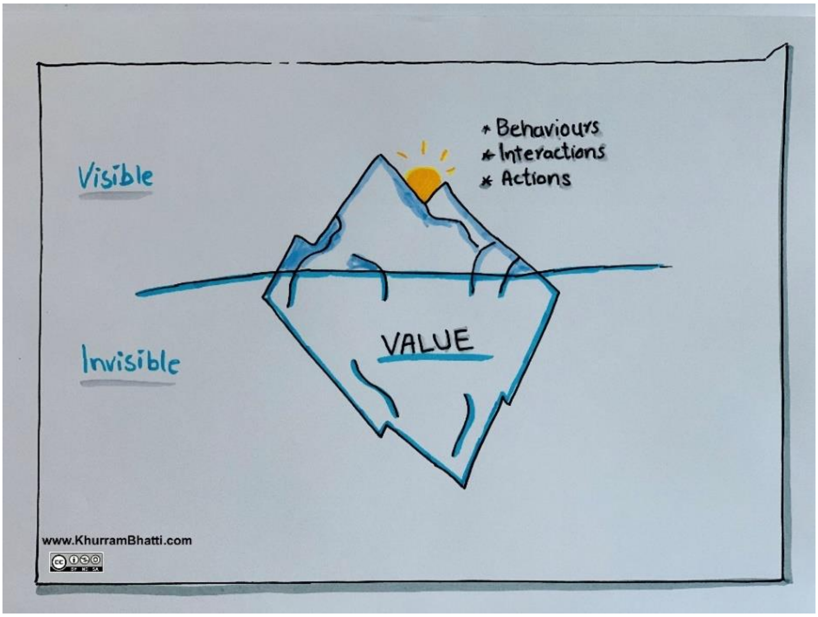
Values become visible when they are connected with demonstrable behaviours, actions, or interactions.

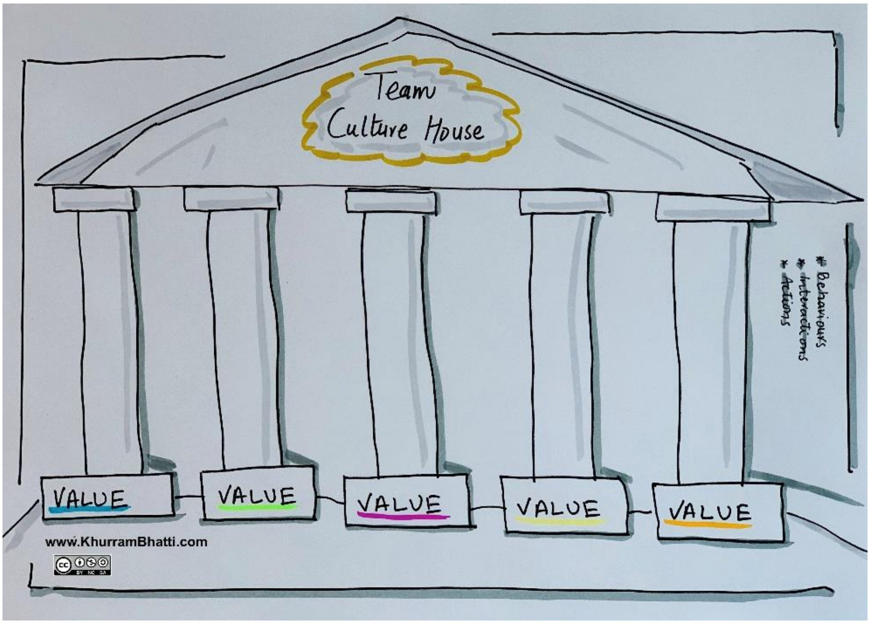
Using the Team Values exercise I have coached teams to design team alliance and working agreements by asking questions from the list below.

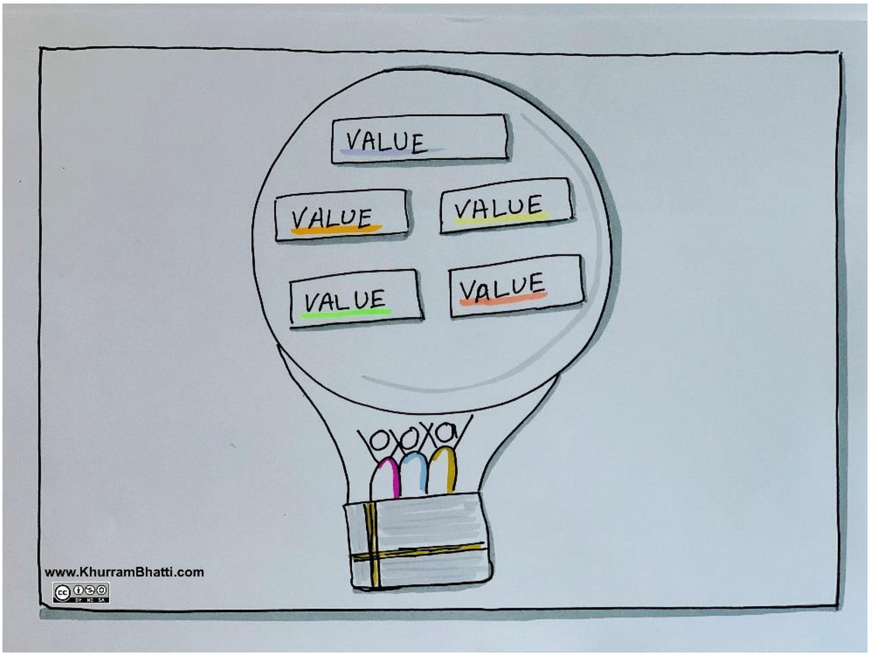
- “What does Value mean to you?”
- “What is the culture, space, or atmosphere you want to create in the team while working remotely? and how do you know you have it?”
- “What kind of culture or climate do you want to create together?”
- “What are the values you want to live by as a team?”
- “How do you want to be with each other?”
- “What will work best in the interactions?”
- “What do they want for team culture and their relationships?”
- “What values they want the team culture to express?”
- “What can you count on when it gets difficult?”
- “Who do you want to be together?”
- “How do you want it to feel?”

Facilitation Guide

For this exercise you'll need to *download the Khurram Value deck* and print it. You can also use the digital version if you wish to.







1. Check-in with the team how everyone is doing *(5 mins)*
2. Explain to the team this is a co-creating activity and everyone's participation is required (unless someone wants to totally opt out)

3. You can use any of the above metaphors to visualise the concept.
4. Invite the team to go through all the values and select top 5 values those resonates to co-create a healthy team culture. *(35-40 mins)*
    - Team can group the values and pick the top one
    - Observe how the team is collaborating and doing the decision making. This will help to understand the team dynamics
5. Invite the team to create a gesture or visualisation of each value *(15-20 mins)*
6. Ask the team what made them select a particular value and how it will contribute to healthy team culture, try to explore the thought process. *(5-10 mins)*
7. Invite the team to define the demonstrable behaviours, actions or interactions for each top 5 values. *(25-30 mins)*
    - Notice it is more actionable rather than fancy words or explanation
8. Invite the team to have a conversation about commitment and accountability to this co-created team culture plan. *(5 mins)*
9. Decide a date to revisit these team values and adapt it based on new information and learnings. *(5 mins)*

Experiment with it and please share your comments and feedback. Thanks.

Inspirations & sources

- ORSC coaching model
- Co-Active coaching model

- *Ourworldindata.com*
- *In an 8-Hour Day, the Average Worker Is Productive for This Many Hours*
- *US Bureau of labor statistics*

About Khurram Bhatti

Khurram's work is about coaching individuals, teams, and organisations to discover better ways in the world of work.

He is an Organisation and Relationship Systems Coaching (ORSC) trained coach, TBR trainer and ICAgile authorized instructor for the agile coaching track. He is an experienced agile coach, facilitator, and agile trainer with more than 17 years of experience in the software industry. He is a founding member of the Pakistan Agile Scrum Exchange (PASE) to spread awareness of agile in Pakistan since 2017.

Khurram enjoys trekking, reading, and playing table tennis!

Ways to Contact Khurram

www.khurrambhatti.com
khurram.bhatti1@gmail.com
www.linkedin.com/in/khurrambhatti/
twitter.com/KhurramBhatti

Stop Sub-Optimizing. Invite — Don't Impose Scrum on Teams.

By Leise Passer Jensen

Sub-optimizing is becoming an excuse for not reaching your company's potential. And imposition kills motivation.

Suddenly 'agile' boomed in the industry. It's only a few years back that 'every' company went agile.

Or so they said. The companies. Their managers.

Photo By Cailtlin Wynne on Unsplash

Boom! We're Agile.

They believed it. But hadn't yet noticed the black smoke.

Maybe companies trust they're agile because they have trained a lot of teams in Scrum or other frameworks. But this is clearly not enough.

One of my highly esteemed mentors recently reminded us that training is for dogs, but that the real goal for Scrum is to educate people with something beyond a Pavlovian response. [1]. What was that again?

…the real goal for Scrum is to educate people with something beyond a Pavlovian response. [1]

My mentor knows I am also a dog trainer. I acknowledged the deepness of his advice.

You don't 'Go Agile' overnight, just like that, by training the teams like a dog. The problem is that many companies still don't realize that training is far from enough.

Can you help fix our problems with Scrum? A potential client asked.

Me: "What problems?"

"Well, our teams still don't deliver what they have committed to. They do work in sprints."

Sometimes, by asking a few questions, we can hint at whether they will achieve what they aim for. For example in this case: Are your teams *cross-functional*? May they *self-organize* and stay *autonomous*? Was agile ever *imposed*?

Q1. Are your teams cross-functional?

Client: "Well, we're kind of semi cross-functional.

The nature of what we do and the type of internal products we deliver do not allow our teams to be fully cross-functional. We don't have enough people. We need to keep the original back-end teams as-is. The UX'ers wish to sit together to feel connected and exchange knowledge daily. Our front-end developers are still inexperienced. We cannot split their team yet since they are all learning from the same mentor. And the DevOps team is too small to merge into the other teams."

Me: "Okay — then go and fix that first!"

"We can work on it together", I added, "engage, and involve everyone on what and how to change. But you have way *too many hand-overs* which slow down everything. That should be fixed." [2]

Client: "What do you mean? I told you we can't change that."

Me: "Hm… but then you just won't reach your potential with Scrum, and should consider doing something else."

Photo by Beate Bachmann on Pixabay

Q2. Are your teams self-organizing and autonomous?

Client: “Well, we’re somewhere between self-organizing and autonomous.

We work in a highly regulated industry. We (managers) believe the teams need direction and control due to that, and we will not burden them with too much responsibility. Thus they must comply with the good- and best practices described in our company policy. We have given them a strict way of reporting progress to help their workday become more efficient. But they are free to do Daily Scrums up to 3 times a week.”

Me: “Okay — go and fix that first!”

I continued: “We have moved from the Industrial Revolution into a world of high complexity and uncertainty. In that type of domain, we can’t fix problems with policies. Your company works with complex adaptive systems: You have to take decisions without knowing all details. Neither best practices nor good practices fit your situation.[3]

We can work on your challenges together. Experiment, engage, and involve everyone on what and how to change. But you are too strict on policies, and your level of control has to be fixed first.

People need to feel *they* are in control so that *they* can feel responsible for *how* they work." [4], [5].

Client: "What do you mean? I told you we are in a highly regulated industry. We can't go away from our as-is policies. They were pretty expensive to create."

Me: "Hm… but then you won't reach your potential with Scrum, and should re-consider if you can benefit enough from it."

Photo by Anja on Pixabay

Q3. Was agile or Scrum imposed on the teams?

Client: "Well, it was a corporate decision, so we had no other choice than mandate Scrum on our teams. "

"Therefore some world champions in Agile Transformations were brought in, told us what and how to do, documented the new policies, and trained the teams to become Scrum Teams."

Me: "I see — I suggest you go and fix that first!"

Client: "That's too late. It's done and over with."

I added: "Well, there are ways to invite, engage, and involve people [6]. That is never too late. Unleash the intrinsic motivation of teams and individuals, and magic can happen. Continual Improvement is one of the cornerstones of Scrum. We can bring it in play, but it requires an effort from all affected parties. Not only the teams. But management as well.

Only then I will be able to help you. And work with everyone interested, so that we can grow your Scrum."

As Mike Cohn once said [7]:

Changing practices is one thing; changing minds is quite another"

Later…

The manager came back to me a few weeks later. " I believed you could tell us how to fix our problems, but I realize now that we need to fix them ourselves. In small steps.

Can you tell me more about this Invite, Engage, and Involve people approach?"

And so we started by making his management meetings optional. We noticed how everyone showed up to his first meeting because they were afraid of what would happen if they didn't. At his next meeting, however, they felt more secure. Some decided to spend time on something else which added more value for them and the company. No penalty. No blame.

Photo by Gerd Altmann on Pixabay

This way they had experienced a new beginning in the management team. One day they could even rightfully call themselves Leadership. Slowly converting fear to trust. In small steps.

Soon we started working with engaging and involving the Scrum Teams.

Conclusion: Stop sub-optimizing! Get into the Spirit

Experiment and hypothesize to fix the real problems in due time. Face reality, but don't make it an excuse for not being able to improve your Scrum adoption.

Learn about the true spirit of Scrum.

Invite — never enforce. Trust people and find ways to involve everyone. [6]

Then you will grow your Scrum. No matter what.

But stop sub-optimizing! And stop imposing Scrum and agile on teams!

References

[1] Coplien, James. “Scrum Patterns Course.” Copenhagen, 9–10 March 2020.

[2] Sutherland, Jeff & Coplien, James. Scrum Pattern: Cross-functional Team, *http://scrumbook.org/product-organization-pattern-language/development-team/cross-functional-team.html, accessed 20-Sep-2020*

[3] Snowden, Dave. “The Cynefin Framework.” By Cognitive Edge, 11. July 2010: *https://youtu.be/N7oz366X0-8*

[4] Sutherland, Jeff & Coplien, James. Scrum Pattern: Self-organizing Team, *http://scrumbook.org/product-organization-pattern-language/development-team/self-organizing-team.html*, accessed 20-Sep-2020

[5] Sutherland, Jeff & Coplien, James. Scrum Pattern: Autonomous Team, “A Scrum Book, The Spirit of the Game” 2019. Scrum Pattern: Autonomous Team, accessed 20-Sep-2020

[6] Mezick, Daniel & Mark Sheffield. “Inviting Leadership. Invitation-Based Change in the New World of Work”, 2018

[7] Cohn, Mike, *https://www.goodreads.com/quotes/7478844-changing-practices-is-one-thing-changing-minds-is-quite-another*, accessed 22-Sep-2020

This article was originally published at
https://baa.tco.ac/3EvL

About Leise Passer Jensen

Leise Passer Jensen is an independent agile coach and trainer. Her background is within IT and systems development, and her strength and passion is to support organisations in becoming places where people really want to work and where the organisational structure scaffolds and enhances the value created. She holds an M.Sc. in Computer Science and English from the University of Copenhagen.

Leise is a skillful communicator of wise principles of leadership. She has a respectful approach to coaching individuals and organizations and takes time to understand frameworks and dynamics of a business.

Ways to Contact Leise

agilit.dk/publications

leise@agilit.dk

www.linkedin.com/in/leisepasserjensen

twitter.com/leisepasser

How to Decline a Stakeholder's Request Without Pissing Them Off

By Lutz Mueller

By the end of this article, you can decline requests from stakeholders to keep your backlog clean and manageable. Your stakeholders will understand why you reject their requests without being annoyed or pissed off. That will strengthen your role as a Product Owner, and you gain respect and trust.

Why You Sometimes Have to Turn Down Requests

Do you get a lot of requests from your stakeholders and customers? A new idea here, which could increase the ROI of your product, a must-have feature there to stand out from the competition, or maybe a change request to existing functionality.

As a Product Owner, you get a lot of requests from your stakeholders. Sometimes you don't have the time to discuss those requests with your stakeholders, so you add new items to your Product Backlog (mostly as titles only) to satisfy your stakeholders and get rid of them for now.

Your product backlog is growing quite fast, and you are looking at a backlog of a few hundred items. Maintaining such a vast backlog takes a lot of time. But this is not the only problem you have: Since an item is on the backlog, the customer/stakeholder expectations are that they will be implemented in the future. That's their logical fallacy, but hard to debunk.

Which Requests Should You Decline?

When you decline a request, your stakeholders or customers are not amused. Maybe they are disappointed, angry, and frustrated. That's normal. That's part of the business.

If you want to gain your stakeholder's trust and respect, you need to have clear criteria for adding an item to your backlog and when not. That helps you survive political discussions, and you do not make yourself vulnerable as a Product Owner.

I recommend declining requests for features which

- do not match the product's vision
- do not support your next strategic step
- do not support achieving OKRs (if you use OKRs)
- are not necessary for the next 3-6 month

A product backlog is an ordered list of requirements for any changes to be MADE to the product.

Three to six months is a time frame you can predict relatively reliable on what to develop and what not.

The further a feature is in the future, the more difficult it is to be sure you will create this feature, as things can change quickly. If so, you need to adjust your roadmap and next steps for your product.

Imagine you added a feature to get rid of the stakeholder because you are very occupied. If you need to adjust your roadmap and next steps and the team will not develop the requested feature, you will have a hard time convincing the stakeholder that it is necessary to cancel their request.

Remember: Once you added a request to your backlog, stakeholders expect them to be implemented in the future.

Take This Into Account Before Declining

Before declining a request, try to take the view of the stakeholder:

- Why is he/she requesting a feature?
- What motivates him/her to ask for the feature?

The most important question before declining a request to yourself is: "Do I have all the information to decide to decline the request?"

Sometimes stakeholders have a more detailed view and insights about a new feature than you have—no problem with that. Suppose you feel you do not have enough insights, set up a short meeting with the stakeholder to gain insights and knowledge about your product's possible future change.

Finally, before talking to the stakeholder and communicating your decision, you need to identify: Is it a "no", or is it a "no, not now".

How to Say No to a Stakeholder's Request

You got a request, collected all the necessary data, thought about it, and finally, you are convinced to reject the request. The worst thing you could do is reject the request without giving a context: "We can not do this feature. It is not a priority". Boom, that's rude. Why?

Try to take the stakeholder's view: Maybe the stakeholder invested a lot of time to provide all the data to convince you to add this feature to your Product Backlog. How would the stakeholder feel? How would the stakeholder behave in the future?

Instead, take the time to talk to the stakeholder. Stakeholders, like every person, want to be heard.

First, express appreciation and thank the stakeholder for the interest in improving the product. If possible, offer an alternative that will produce the same outcome but with a different approach. Usually, stakeholders are interested in specific results, f.e. getting new users, reducing churn, etc. If you can explain how you will achieve the desired outcome with a different approach or feature, they will agree on your rejection more easily.

"No, Not Now"

If you reject the request for now, but not for the future, give the request back to the stakeholder and tell them when they can come back to you. You hand the responsibility for this request back to the stakeholder. Don't create a ticket in your backlog for this request. If this request is still vital in the future, the stakeholder will come back to you with this request.

"No"

Saying "no" to somebody is not easy. You know you will disappoint someone. Maybe this person will express their disappointment with facial expressions and gestures. Some people will get angry; some will take it personally.

There is nothing you can do about it. That's part of the game. But there is something you can do to strengthen your role as Product Owner:

Offer one strong reason why you decline a request. Just one reason. If you offer various reasons, some stakeholders will pick the weakest reason and argue against this reason. Support your strong reason by market data, user behavior, and other product-related data. Please don't make it personal.

How to Communicate

The way you communicate the rejection is the key to cooperation on an equal footing with the stakeholder. Avoid using "I", instead use "we" language:

Don't: "I don't think we should focus on that feature now."

Do: "Our research told us that…"

Combined with a strong reason, "we" language will help you as a Product Owner to convince the stakeholder that rejecting a given feature is the correct choice.

What, if a Stakeholder Still Insists on a Feature

Sometimes you will face a situation where all the tips and tricks above will not work. You may have a stakeholder who still insists on a feature.

As your backlog should only contain items for a given time frame (*read here why*), f.e. 3-6 months, you can only accept a request by removing an existing item from your backlog. Ask the stakeholder which other features you should remove to have the capacity to develop the requested feature.

If you have features from different stakeholders in your backlog and have a hard time convincing a stakeholder not to develop a requested feature, try to get them all on the table and discuss the implications and consequences of a given request.

This article was originally published at
https://baa.tco.ac/3El-

About Lutz Mueller

I help companies to bring their products to market faster through agile product development.

By teaching and applying agile development methods and techniques, I enable my customers to develop products in shorter development cycles with faster customer feedback.

Together we implement product metrics to validate business hypotheses and develop products that customers really want.

By creating a continuous improvement environment with supportive metrics, my clients get a robust feedback mechanism to increase flow, decrease latency, and build a learning culture.

Ways to Contact Lutz

www.lutz-mueller.com

info@lutz-mueller.com

www.linkedin.com/in/lutz-mueller/

twitter.com/DerLutzMueller

Metaphorical thinking (Using plants, eels, birds for reflections!)

By Madhavi Ledalla

If you want your children to be smart, tell them stories. If you want them to be brilliant, tell them more stories.- Albert Einstein

Story telling using metaphors has proven to be instrumental in creating an impact along with long lasting memories. If you are keen to know more about metaphors then here is a post for you.

Analogies are based on stories and metaphors are built on these analogies. Try narrating personal encounters using metaphors and experience how the audience can relate to your stories.

Metaphors offer exceptional ways to communicate ideas naturally. The crux lies in selecting an appropriate metaphor that resonates with your audience. Metaphors invoke creative part of brain that is stimulated by visual illustrations.

Fostering creative ideation using metaphors is called as metaphorical thinking and this way of stimulating the thinking process helps analyze and decode any complex situations at ease and brings a new perspective into the forefront. Reinforcement of this concept is that metaphors are being used in almost all social media promotions.

Due to the fact the human brains can relate to analogies and visual illustrations, metaphors prove to be very useful while facilitating conversations and foster an "outside the box" thinking. Using story telling as a medium through metaphors opens up new perspectives and helps the audience connect to the example and thus they will be able to visualize and paint a picture of the reality at ease.

I could see a significant improvement in the involvement levels of my team members when I used stories from my personal encounters related to natural things that happen around us like flying a kite, climbing a mountain, riding a bicycle etc. These metaphors helped me create a light weight structure to drive the conversations and led to the emergence of insightful outcomes.

The metaphor as a tool is a powerful hook for our imaginations to latch on to and is an effective way to trigger new behavior. Metaphors ease the cognitive strain of our brain and the brain responds effectively to visual elements as it finds easier to assimilate the new learning and thus the knowledge acquired stays far more firmly in the memory.

The benefits I see using these metaphors are-

- Help in lateral thinking
- Help describe our thought process better
- Leads to greater engagement
- Aid in connecting to the concepts in a more natural form
- Help in explaining a complex problem effectively
- Create enthusiasm and improves the listening skills as audience start connecting to the metaphors

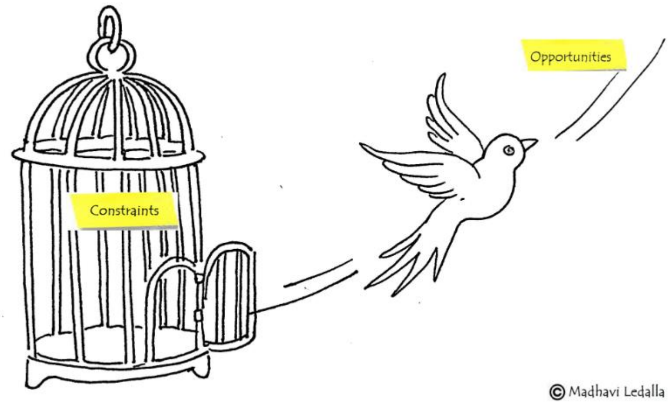
- Act as influential hooks for the writers and aids in unlocking the passive potential using their own imagination and thoughts

Discussion using metaphor is like storytelling and we all know that storytelling has the greatest impact, proof of this is that all of us cherish and remember our childhood stories.

In coaching conversations, it is believed that everyone is naturally creative, resourceful, and whole. Being aware of how you've been doing and what you're doing in any situation is vital for reflection. Innovation is the key and one needs to cultivate this mindset. So try using metaphors and quote real-life situations as you see fit while helping individuals and teams to reflect on their present. Visual metaphors can be used to spark the innovation capabilities as well. For example, I use "setting the bird free from the cage" to discuss the current constraints the team has to explore the unlimited opportunities in front of them which is similar to a bird not being able to explore its world due to it being confined to the cage.

Here is an example of how I use "Set me Free" analogy

Sometimes, teams work across countries & continents. All individuals cannot travel across the globe to meet everyone in their team. This lack of personal interaction could be a constraint that is placing a roadblock in collaboration. How can you convert these constraints into opportunities?



Set me Free!

In one of my retrospective meetings, I described the situation of a bird that is confined to a cage and asked the team what could be done to let the bird explore the world. “Set it free! Let it fly,” came the prompt response.

I used that response and asked the team, “Do we think our team also has similar constraints because of which we cannot explore the limitless possibilities in front of us?” For a moment the team was silent, but they slowly opened up. This was an aha moment for me and I understood how well the team could relate to setting a bird free.

I repeatedly say that certain occurrences around us could connect to the team in their retrospectives. This method is especially useful for teams and leadership to debate on constraints they face and ways to convert these into opportunities.

Instructions

- Draw the picture above (or print it out) on the whiteboard.
- Describe the constraints the bird feels in a cage, ask the

team how it feels when it is set free.

- Ask the team to think of constraints they face in their teams that is limiting them from exploring the world of opportunities. Ask them to write it down on post-its and put it inside the cage.

- Ask them about the opportunities they see in front of them they would like to use if set free from the constraints. Write these down on post-its and put it outside the cage.

- Encourage the team to think through the limitations and provide leadership an insight into the constraints the team faces and how leadership support could help the team explore the world of possibilities. As part of this discussion, also ask them to think about what they can do to overcome a few of these constraints. In effect, this exercise should help the team arrive at what is their scope of control.

The "Set Me Free" metaphor can also be used in coaching conversations where you would like your client to think of the opportunities they have in front of them and also vent out the constraints they are surrounded by and then generate a discussion of how can they overcome the constrains and meet their goals.

Avoid overusing metaphors and use something that your audience can connect to. Remember a metaphor should aid in understanding and clarity. Use examples that readers can relate to and connect easily, thus I would recommend to use examples from nature as far as possible. Also please remember to consider cultural aspect while quoting metaphors as different cultures can perceive differently.

If you are interested in exploring metaphors, do read my book – "Retrospectives for Everyone"

Would be glad to hear your experiences of using metaphors!

This article was originally published at
https://baa.tco.ac/3Ew-

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.

She works with leadership & teams to guide them through transformation. She is a Speaker, Reviewer & Organizer in Regional and Global Agile conferences. She is the author of the book - "Retrospectives for Everyone."

Ways to Contact Madhavi

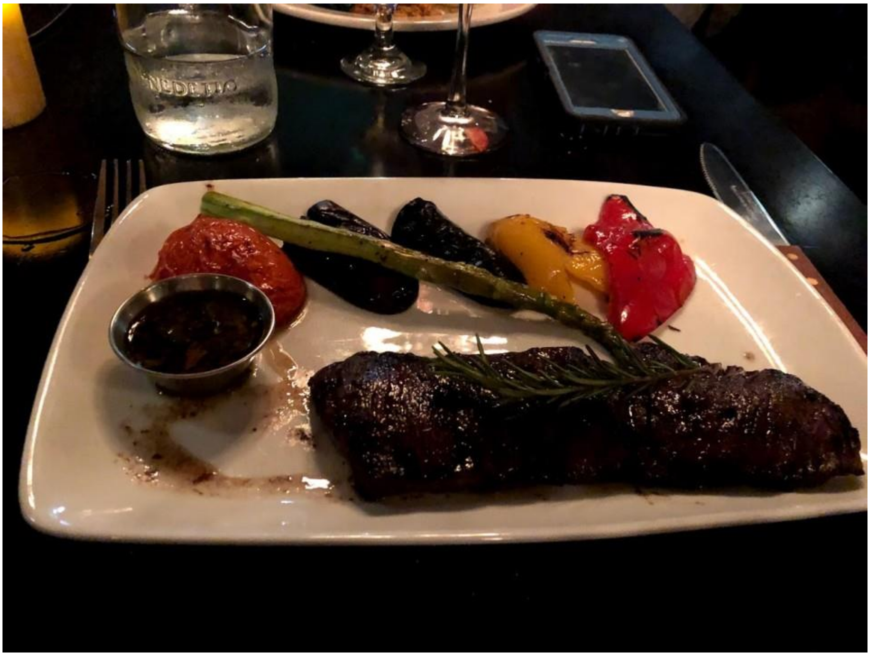
lmadhavi.wordpress.com

lmadhavi@gmail.com

www.linkedin.com/in/madhavi-ledalla-6454237/

One Answer That Is Never Wrong

By Mariya Breyter



Which Agile transformation framework to pick and what is the right metrics to use?

In times of disruption, times of change, like the one we are going through now, there is a lot of questions: what does it mean to each of us? what does it mean to the world? how do we cope? navigate? leave it better that we went into it? What is the new state and what did we learn?

As we go through the multiple questions, there is a lesson that my consulting experience taught me. There is one answer that is never wrong: *it depends*.

As an agile practitioner, conference speaker, and organizational change coach, I get a lot of questions from my network: how do we execute an Agile transformation at an enterprise level? Which framework to pick? What should we measure? How should we report progress?

For nine enterprise Agile transformations I went through, there is no single answer. There is a pattern though that unites those that have been successful. It is not any specific framework, we used multiple and invented one on our own.

There are multiple articles and comparisons of *Scrum at Scale, Scaled Agile Framework* (SAFe), *Large Scale Scrum* (Less), *Disciplined Agile Delivery* (DAD), even *Spotify org structure* is sometimes considered an Agile scaling model.

Once the core framework or a custom approach is defined, the next question comes up: *is there a single set of metrics to use?* There is a lot of evidence that actionable metrics is important while vanity metrics is misinforming. No one would argue nowadays in favor of vanity metrics, such as productivity data. The concept of outcomes vs. outputs is no longer a point of discussion. However, what type of metrics is considered actionable, and why? Quality metrics or predictability metrics? Feature performance or user feedback? Leading or lagging indicators? Execution metrics or market data? Web analytics or sales reports?

At each level, from the enterprise level to its value stream/portfolios, and all the way to individual teams, each organization makes its own choices what to *measure*: employee engagement, customer satisfaction, continuous improvement, time-to-market, quality, progress against OKRs, any other delivery metrics?

My answer to each of these questions remains the same: *it depends.* If the organizational goal is to improve the quality of their product, they measure quality, if business performance is the key, they measure RoI, if customer satisfaction is their top concern, they measure customer happiness or Net Promoter Score. Measuring everything means measuring nothing. The approach is to set measurable goals (OKRs) and measure key results based on relevant data points – and once the key results are met, the next step is to move to the next high-performance improvement, the next high priority objective with its own key results, or to set up a new one.

The same approach applies to the *framework selection:* it depends on organizational culture, size, openness to change – it's never "one size fits all", and no transformation is exactly the same as another one. Then, the question is: are there any common ways of defining this "fit for purpose" in an organizational transformation? Of course, it depends. Some of the items to consider are organizational readiness for change, leadership support, clear organizational objectives, level of risk tolerance, level of commitment, and sponsorship of the effort. Success will be defined by commitment, leadership support, open mind, and clear objectives. The human factor of organizational transformation is the most important one: is the organization committed to change? is there psychological safety or people are afraid of losing their jobs as a result of this change? Are the objectives clear and inspirational? Are expectations realistic? Is there an explicit and measurable alignment on outcomes?

Agile transformation is like a meal: is there a perfect meal? It depends. Is it a meal for me or for you? For an adult or for a child? Morning meal or afternoon meal? Which season? It depends. My preference would be different from yours. My preference today would be different from five years ago or five years from now. And even that is not enough. How many times have you ordered a meal in a restaurant and did not like it? And maybe it was not a bad meal – it was just not what you are used to or not what you expected. It simply did not fit your need or did not meet your expectations.

Agile transformations are similar. There are general pre-requisites: clear objectives, psychological safety, leadership support, and others mentioned above. If these pre-requisites are in place, there is no doubt that the organization will be able to find the path to their transformation – through trial and error, continuous improvement, and ongoing feedback loop, no matter which Agile framework they put in place. At the end of the day, Agile transformation *is not about processes, it's all about people*.

This article was originally published at
https://baa.tco.ac/3Egc

About Mariya Breyter

Mariya Breyter brings over 20 years of leadership experience to the Agile and Lean community. Her passion for creating high-performance teams and delivering superior products to clients through efficient, lean processes had produced success after success in companies ranging from Fortune 100 insurance and financial services firms to educational and media businesses.

Mariya has a Ph.D. in Computational Linguistics followed by a Post-Doctorate at Stanford University. At home, Mariya is using lean principles to organize family activities, while her kids use Kanban to achieve inspirational OKRs.

Ways to Contact Mariya

www.agileleantransformation.com

maria_breiter@yahoo.com

www.linkedin.com/in/mariyabreyter

twitter.com/mbreyter

Scrum Anti-Patterns: Large Product Backlog

By Mark Levison

A design pattern is a description of a solution to a recurring problem. It outlines the elements that are necessary to solve the challenge without prompting the reader to address the issue in a specific way.

Unfortunately, we also regularly see recurring patterns of ineffective behaviour. These are called anti-patterns. They get your hopes up, then leave you with a bigger mess to clean up than what you started with. Learning how to spot anti-patterns will save you time, frustration, and antacid medication.

The following is an exploration of one of the most common anti-patterns.

Anti-Pattern1 Name: Large Product Backlog

Aliases: Wishlist, Too Many Promises, Electronic Backlog Tool (e.g. JIRA, Target Process, etc.)

Scale: Team and across multiple teams

Related Anti-Patterns: Product Backlog Refinement goes on forever, Product Backlog used as a Parking lot; Feature Factory...

Potential Solutions:

- Don't record every suggestion from the client
- Say "No" and "Not Now"
- Search for older Backlog Items and delete them
- Use a prioritization model that helps you make clear decisions about what is not included in the product

Why a Large Product Backlog Might Seem Like a Good Idea

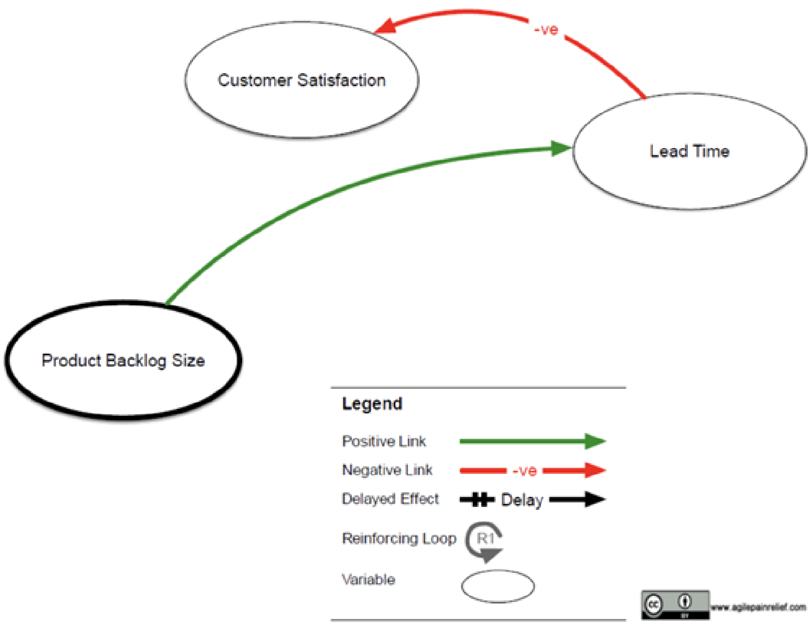
Product owners (POs) new to *Scrum* are eager to please the customer, the end-user, and every stakeholder in their universe. They get asked for a feature, user story, or requirement (which hereafter I will refer to as a Product Backlog Item, or simply 'item'). The person asking explains why they think it's important and makes it clear that it must be added to the product backlog. The new PO adds it to the bottom of an ever-growing list.

At first, this works well and the customer/end-user/stakeholder feels satisfied because their item has been added to the Product Backlog and they will soon get their feature. However, a problem is brewing just underneath the surface of the situation. Over time, the Product Backlog grows from a manageable 50 to 60 items to 200 to 300. In some cases, I've seen clients with backlogs over 500 items long. Our brains just haven't evolved to deal with and comprehend such long lists.

To illustrate this problem, I'm going to start by making a few assumptions: All of the items in the product backlog are of equal size, and a team averages seven to 10 items per sprint.2 Let's also assume two-week sprints. Arithmetic tells us that, to finish a list of 200 items, it's going to take 20 to 25 sprints, or most of a year. In practice, it's usually worse than this, because items lower in priority tend to be larger in size, so a 200-item product backlog might likely be over a year of work.

Imagine telling your customer, "If we add that item to the bottom of the product backlog, it will take eight to 10 months to finish." That isn't likely to delight them. However, simply adding it to the backlog without first discussing it in more detail, then having it be forgotten isn't going to make them happy either. Customers can easily create wish lists, but the product owner needs to provide information and context that they don't have, along with a healthy dose of realism, so that their perspective and expectations are attainable. Maintaining the product backlog at a manageable size ensures that.

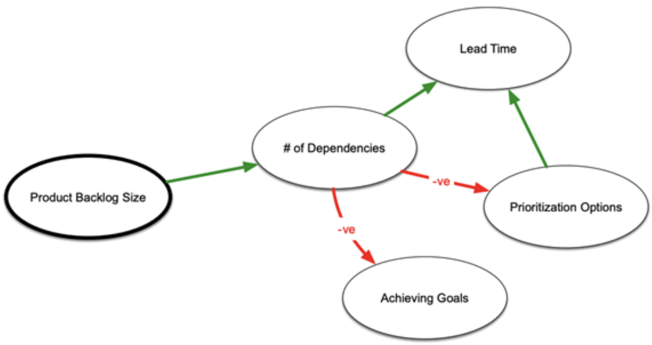
Consequences of Using Large Product Backlogs



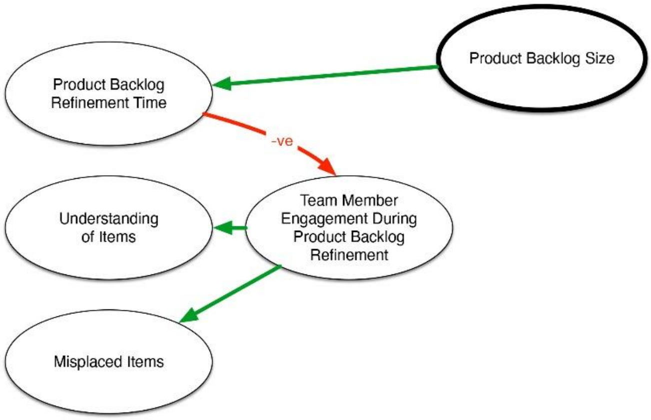
As the product backlog grows larger:

- The lead time[3] grows. As the lead time grows, the customer becomes dissatisfied, wondering why their feature is taking so long. Imagine lead time like ordering a car from a manufacturer — how long from the moment you place the order to when it's delivered. Think about what it does to customer satisfaction if they must wait eight to 10 months for their new car.

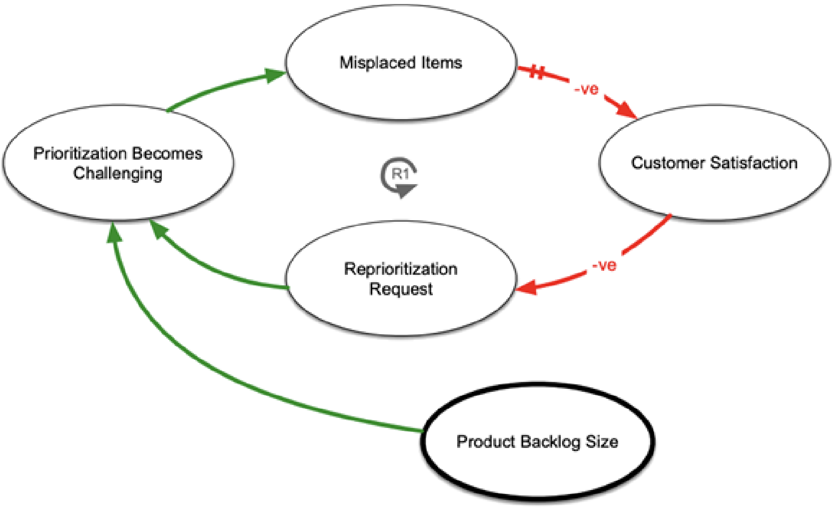
- The number of dependencies[4] also increases lead time by creating delays. In a multi-team situation, dependencies between teams increase the likelihood of missing the target/goal. The number of dependencies also reduces the number of prioritization options. As the number of prioritization options is reduced, the lead time increases. As the number of dependencies increases, the likelihood of achieving goals decreases.



- Product backlog refinement takes longer. The longer the list, the more time it takes to review, comprehend, and understand. As product backlog refinement takes longer, team members become less engaged. After all, who enjoys an activity that consumes more time with every sprint? As they become disengaged, they have a reduced understanding of the product backlog Items and are more likely to build a version of an item differently from what the customer wanted. Finally, as product backlog refinement takes longer, it is more likely that the team will misplace items that have become buried.

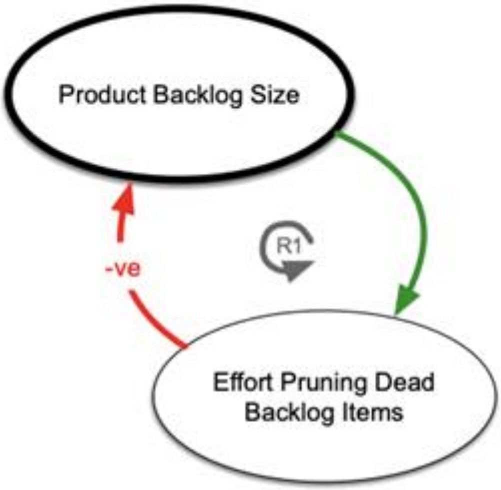


- Prioritization becomes challenging. It is already difficult

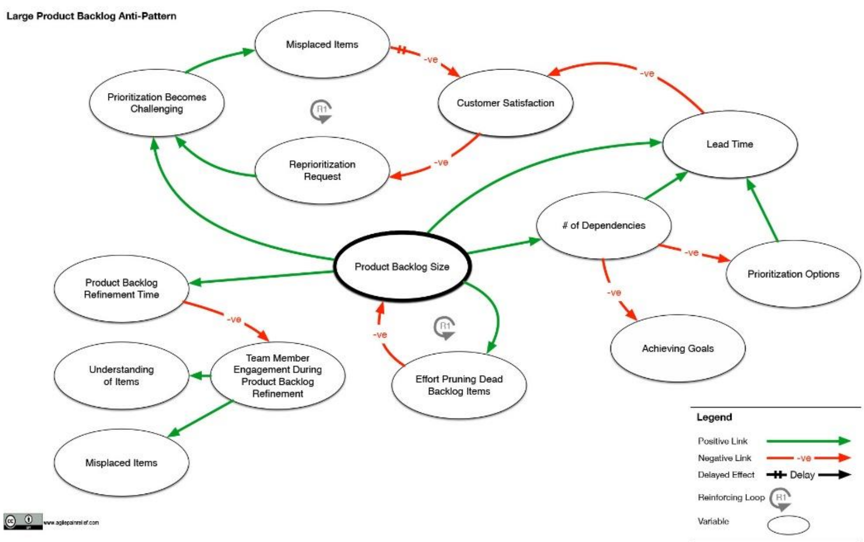


to do an effective job of prioritizing a list of 50 to 60 items, let alone anything larger. As you struggle to prioritize, items get misplaced in the product backlog and the customer grows frustrated, asking, "Where is my item? Why isn't my item a priority?" Usually, they will then ask for their item to be made a priority (prioritization request) and, if you grant the request, it will increase the lead time on an item that someone else felt was a priority.

- Pruning dead items becomes harder. As the product backlog grows larger, it becomes harder to spot the dead items (i.e. items that are no longer relevant). As dead backlog Items get missed, the product backlog grows ever larger in size. This is called a negatively reinforcing feedback loop.



Here are those effects all in one illustration:



Typical Causes

- Absent product owner: The product owner has the job title, however, they put little time into performing the role
- Product owner by committee: For example, four or five people think they're the PO. They form a committee and just keep adding to the backlog
- Product owner fails to say no to customers and stakeholders or isn't empowered to say no
- Saying yes to a new item without removing an existing item from the product backlog
- Never pruning older items
- Infrequent releases (less often than every three months) lead to unmet needs piling up and customers desperate to get their feature in the next release
- Same item appearing in the product backlog multiple times, each phrased in a slightly different way, each with different supporting information
- Failure to fix bugs when they're found. Instead, adding them to the product backlog
- Misuse of product backlog management tools. It's easy to make really long lists that can't be maintained. Made worse in that some tools offer, as a feature, a way of hiding small user stories as children of a bigger story
- Lack of portfolio management (in a multi-team situation)
- Failure to appreciate that there is no value in breaking down items for the longer term (months down the road) until it is closer to the time to do them
- Executing on an existing predetermined plan vs. discovering the system the client needs. Some people believe that if you're replacing a legacy system all we do is copy all the elements of the existing system into a new one. As anyone who has done this before can tell

you, the needs of the users will have changed a great deal since the legacy system was developed. So even when replacing an application, we still need to spend time discovering what it's really needed for.

Solutions

- Measure lead time
- A tool that fades the older items in the product backlog, whereby if they haven't been started after three to four months, they are deleted
- Don't break down product backlog Items that won't be worked on for the next three sprints. If there are small items more than three sprints away, collapse them into larger items, then delete/archive
- Product owner learns to say no, perhaps by saying "not now" or "If I say yes, it will likely take four or more months to finish. If it's still important in three months, please ask again."
- Every time a new item is added, remove a less important item
- In a multi-client/stakeholder situation, limit each party to four to five items in the queue at one time. To add a new one, an existing one needs to be either finished or deleted because it is no longer as important
- Tease apart the existing mess with a story map and then throw away most of the now-extraneous details. Use the major items (often called major tasks) at the top of the story map as placeholders. If an item isn't important in the next few months, don't create individual user stories under the placeholders.
- (Rarely) *portfolio management*
- *Find a coach* to help you navigate challenges as a product owner and dysfunctions on your team.
- *Get certified*! Scrum Alliance offers product owner,

ScrumMaster, and Scrum developer certifications and support to help you navigate your journey as an agilist.

Notes

1. An anti-pattern is a common response to a recurring problem that is usually ineffective and risks being highly counterproductive. ~ *Wikipedia*
2. Both of these assumptions are weak, however, the truth just makes the problems worse.
3. *Source*
4. *Source*

Sprint Goals Provide Purpose

By Mark Levison

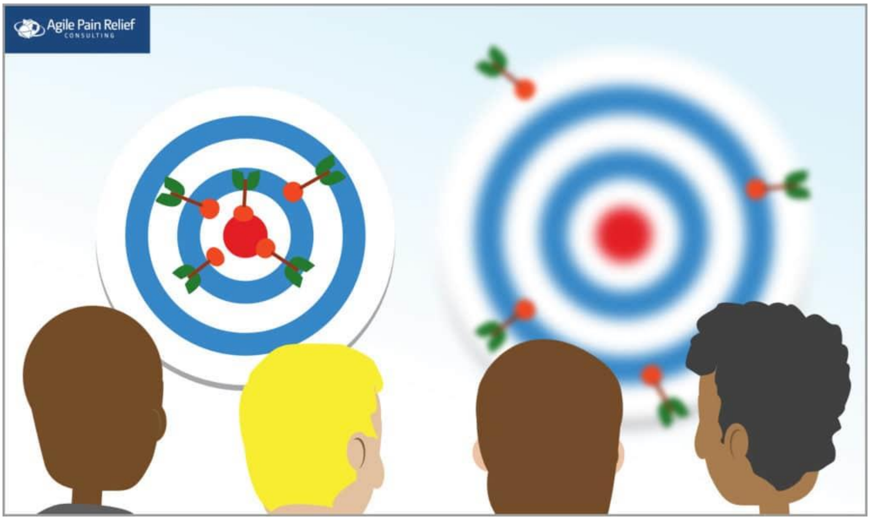


Image attribution: Agile Pain Relief Consulting

Go Beyond Merely Completing Work Lists

A Sprint should be so much more than just completing a number of User Stories or fixing bugs. If your Sprints are merely about ticking off items on a scattered work list without having a shared understanding of why they're important or what purpose they serve, *your Development Team* will struggle to keep focus in the short-term and will take longer to *become highly effective* in the long run.

Research shows that people, whether acting individually or as a team, achieve more when working toward an objective that is specific, challenging, and concrete.[1] For that, we need clear Sprint Goals.

Why Do We Have Sprint Goals?

In Scrum, we ask a Team to undertake Sprint Planning to agree on what they will achieve in the next Sprint, based on their expected capacity. As an outcome of this, the team should leave their planning session with a clear goal.

This provides stable direction, with flexibility to re-evaluate work, throughout the Sprint. It is the *Why* of a Sprint, establishing purpose and commitment.

By participating in setting a Goal, Team members experience a sense of ownership of it and gain a better understanding of the overall problem. On occasion, it also helps them find better solutions than originally planned because that overall perspective can help see things during the Sprint that aren't obvious if looking only at the individual parts.

The Sprint Goal provides focus in Daily Scrum and an opportunity to refocus if the Sprint goes off-track. Finally, jointly identifying and achieving shared goals is a key element for growing a group of people from a working group to a true team.[2]

How to Make Better Sprint Goals

In my experience, most Sprint Goals are not clear. Some poor examples I've seen are:

- Fix 10 bugs
- Finish 7 unrelated User Stories
- Complete the work assigned to the team in JIRA (*yes, this is remarkably anti-Agile and ineffective, however, I see it all too often*)

None of these examples help focus the Team, nor do they provide clarity on what they're seeking to achieve.

So, what makes a better Sprint Goal? A good Goal answers questions such as: Why is it worthwhile to undertake this Sprint? Are we attempting to solve a problem? Are we implementing a feature or clarifying an assumption?

Improved versions might include:

- Reduce the shopping cart abandon rate from 50% to 30% by improving usability and performance – Solves a problem. We're losing sales because we have a poor checkout experience.
- Add filters to the existing product search results so that buyers spend less time finding items that meet their needs – Implements a feature.
- Offer free shipping for orders over $40 – Tests an assumption that free shipping will increase the amount people spend per transaction.

Should the Product Owner set the Sprint Goal? No. It's good for them to go into Sprint Planning with some business or product objective in mind, but it's through negotiation with the Development Team that the actual Sprint Goal emerges, as all grow towards a shared understanding of what is desirable and achievable in the Sprint. "Achievable" means possible to accomplish while upholding the quality agreed to in the Definition of "Done."

Agile Coach Bob Galen suggests that you imagine crafting an email to invite your whole company to your Sprint Review. What will you put in the subject line and first few sentences to entice them to attend?[3]

That's your clue for your Sprint Goal – the shared understanding created between the Product Owner and the Development Team of the desired outcome of the Sprint.

[1] "*Building a Practically Useful Theory of Goal Setting and Task*

Motivation A 35-Year Odyssey" and *New Developments in Goal Setting and Task Performance*, both by Edwin Locke and Gary Latham

[2] *"High Performance Teams: What the research says"* by David Wilkinson, –The Oxford Review Feb 2019

[3] "*Sprint Goals – Are They Important*" by Robert Galen

About Mark Levison

Mark is now an Author, Certified Scrum Trainer, and Consultant with Agile Pain Relief Consulting.

He has introduced Scrum, Lean, and other Agile methods to a number of organizations including: Government of Canada departments, major financial and insurance institutions, and software companies, as well as individuals across Canada.

He also does consulting work with organizations that are seeking to evolve.

Ways to Contact Mark

agilepainrelief.com
mark@agilepainrelief.com
www.linkedin.com/in/marklevison
twitter.com/mlevison

I Am Your Agile Coach, But I Don't Notice You

By Matt Kirilov

It's time that I break my FOBR (Fear of Being Redundant) streak and write about what's been on my mind for a while.

A few months back my mentor and co-owner of *the leading coaching academy* in the Agile Coaching and coaching worlds, *Cherie Silas* mentioned that *people like being noticed* during one of her advanced coaching classes I was lucky to be invited to. I knew that I was in the right class because everyone there was smarter and more experienced than me. A proof of this was that when I asked about the importance of people being noticed, everyone else seemed to be bored with the subject.

The only reference I had about people being noticed or seen up until that point was informed by a sexist stereotype that often gets rolled into a dating advice - that *women like to be seen.*

Yes, ...and...I like to be seen too, being an introvert and all. And so do you! There is no need for the obligatory neuroscience and psychology citations to back up something that obvious.

It is not that we do not see each other. It is more of a +1 to the common communication anti-patterns - talking past one another without truly listening to each other. I do that too and I suspect that sometimes you do that as well. Oh, you don't? Really?!?! *So you never click the 'Like' button without fully viewing the content, ever?*

I will take a guess that as coaches we work a lot more on our *listening skills* than on our '*noticing people*' skills. Perhaps that is why the *ICF* has:

Notices, acknowledges and explores that client's emotions, energy shifts, non-verbal cues or other behaviors.

and

Integrates the client's words, tone of voice, and body language to determine the full meaning of what is being communicated.

both nested under the *Listens Actively* core competency. The federation came a long way changing the verbal context of *Active Listening* to the more wholistic *Listens Actively* in 2020. I would go even farther and grant the more visual and kinetic competency of Noticing (seeing) its own core competency and name it something like *Active Noticing*.

My recent focus on noticing people has led me onto a new Agile Coaching learning plane. At times I get so tightly wound around my recommendations that I do become an Agile hammer looking for Agile nails to hit. Worse, I temporarily black out on *Norm Kerth*'s *Prime Directive* and on the directive's more evolved version of:

Holding people as naturally creative, resourceful, whole, and capable.

The wise words hold true for me until I walk into a room full of team members who are worlds away from 'doing' my ideal of Agile. All of a sudden my mind flips a switch and I instantly start viewing these people as *broken*.

They practice *broken Agile*, therefore they are *broken people.*

-- my mind instantly concludes. This mindset immediately causes me to show up as judgmental and offensive to the people in that room.

Who is this guy to think that we are all broken? We've kept the lights on in this company for X years.

-- is people's natural reaction. I feel them getting defensive and irritated by the minute, which then farther reinforces my belief that:

Oh, they are getting defensive and reject me and my brilliant Agile stuff. They are definitely broken.

This makes me want to *fix them* (usually disguised as an *urge to help*) even more. Initially professionalism and common courtesy stops people from resisting my 'help.' However, after a short while in our engagement people start doing what people have done for many thousands of years - they give in to what has been keeping them safe - *fight, or freeze, or flight*.

No. People will not freeze in front of nor dash away from an Agile Coach, but they will gradually start to passively ignore my advances towards *fixing* them (no matter how well intended).

Ending in such, albeit all too common, situation breaks an Agile Coach's little heart.

If they only knew how much I care about them and how much I want to help them!

Then I revamp my effort to 'reach' these 'broken' people and 'save' them by offering them so called better ways of working. I pick myself up by my bootstraps and try even harder. On occasion people try things and take on our advice and even succeed with it. Alas, the vast majority of people will push back, as they should. The discipline of *Systems Thinking* has recognized this as *Law#2*:

The harder you push, the harder the system pushes back.

The system is a metaphor for your organization. And since the cubicles, conference rooms, and cafeterias are minor contributors to company strategy the organization is the sum of its people.

You may think that your brain is logical and that by default it makes the right assumptions about people. The proof of the opposite is in the pudding - as an industry us Agile Coaches spend a lot of time and billions of billing hours trying to fix people who are not broken.

What if we fixed ourselves first?

What if we did recognize that people are naturally creative, resourceful, whole, and capable (and therefore, by definition *not broken*)?

How would you show up as an Agile Coach with such an assumption?

How would people show up to you differently if you approached them in that way?

All of this is easy to assume and hard to put into practice because our brains are brilliant at tricking us into assumptions such as the ones that people are broken. It is much easier to stoke our Agile Coach egos and play saviors of the broken team members and leaders, rather than recognize that people are fully capable and that's how they got their gigs in the first place.

Now the harder part.

What if after years of practicing not fixing people we took it a notch up and started *noticing* people?

I am offering the notion to not only *stop trying to fix people who are not broken*, but to assume that they are fully capable and to start *noticing them*. Maybe we can notice them not only for a split second and register it as a piece of information in our brain, but reflect what we notice back to them. Even us Agile Coaches like to be seen and acknowledged when not on our high horses.

How differently are people likely to reach out to you for advice if you approached them in this way?

How does this land with you? Let's have a discussion.

This article was originally published at
https://baa.tco.ac/3ENj

About Matt Kirilov

Matt Kirilov has a well-rounded set of skills that deepen in both the coaching, teaming (team development), and group psychodynamics competencies. He is passionate about leveraging relationships to partner with clients striving for positive change in organizational systems.

Professionalizing Agile Coaching and adopting core coaching principles by the Agile community are very close to Matt's heart. The interdisciplinary intersections are the places where you can find him working on his growth edge.

Ways to Contact Matt

tandemcoaching.academy

mattdkirilov@gmail.com

www.linkedin.com/in/matt-kirilov

Scaling Organisations (1) — Consider Not To

By Matthew Croker

As multiple clients start shopping by and projects start to take shape, organisations get this temptation of increasing their footprint on the market. It is, perhaps, the ecstasy of gold that rushes through the veins of many organizations passing through a very particular phase of their existence: that sweet spot where roping in more people starts looking as the natural next step, a sweet spot that tends to hit organizations most when they reach a headcount of between 100 to 150 people.

Maybe because many organisations passing through this phase are essentially similar to teenagers who give the world a linear interpretation (after all, that is, roughly, what they have experienced up until that point), the complexities of scaling are often missed out. In this series of articles we would like to propose a slightly different point of view when it comes to scaling organisations, and we will start by giving an unusual advice: *start approaching scaling by considering not to scale as a first step.*

It is not a matter of escaping the reality that at the end of the day organisations are bound to *grow* (whether financially, in their product range or headcount), the advice is to delay scaling for as long as it remains feasible.

This series was inspired by Geoffrey West's book *Scale: The Universal Laws of Growth, Innovation, Sustainability, and the Pace of Life in Organisms, Cities, Economies, and Companies* in which the author condenses scaling into three main points:

1. Scaled organisms have space filling networks
2. They have nodes of invariant size
3. They are built to be efficient in their use of energy and resources

Scaling organisations requires an understanding of the organisation's capability to handle the load of scaling. As organisations rope in more and more people the dynamics and foundations are tested, and should the organisation (pretty much like organisms) not have managed to consolidate its networks, nodes and resource efficiency, it will collapse under its own ambitions.

In this article we will address the three points above by using principles and practices from Kanban. Why Kanban? Because Kanban is concerned most with systems and data, and we consider this as the right level of perspective we need for the scope of this article.

Space Filling Networks

Work and workplaces are essentially networks that transfer knowledge and value. When we talk about an organism we talk about networks of veins, of branches. When we talk about cities, we talk about the electricity grid and the interconnected networks of roads. It is because of these networks that the organisms or cities are considered as a *single* unit. When scaling that unit, the network needs to scale too in order to support and sustain the change. Failing to do so, the extension will be detached and the unit will become frail.

When we talk about organisations we should also talk with networks in mind, yet what are the spaces they ought to fill? In *their article about Buurtzorg*, which was also an inspiration for this article, Corporate Rebels mentioned the organization's network spread of cooperative teams across the various geographical locations as an example of space filling networks. The marketplace is one space that an organization as a network aims at filling, and it is an external space vital to its existence that will ultimately justify and sustain its purpose to scale. In the context of this article, the question to ask is "what is the business value that our organisation has not yet untapped, and that we can unlock without the need of increasing our headcount?"

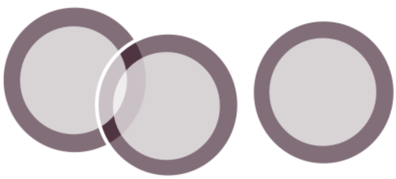
Another space the organisation-as-a-network should aim at filling is the organisation's existing state of being *organised*. Essentially: "How organized is the organisation, really?", the playing with words is intentional.

Behold, for this might inspire for the curse of micro-management for those who rush. Let us, instead, talk Kanban for some lines, change the wording and ask "How does our current flow of delivering value look like?"

Mapping the flow will kick-start the understanding of the processes and types of work that are necessary for the company to deliver its products. In some companies there might be blind-spots, in other companies there might be bureaucratic cycles that tangle the network. The exercise of listing down the steps will make all of these visible, and visibility is an essential first when it comes to understanding.

So many other questions can be asked once the process is visible. Can you spot the waiting phases? Would a scaled version of your organisation survive with your current flows? Order will guarantee better chances of a successful effort for scaling.

Nodes of Invariant Sizes



Organisms have cells and capillaries that cannot be too big or too small for them to serve their purpose. Organisations too have these nodes, and their nodes will also be crucial in understanding the organization's readiness for scaling.

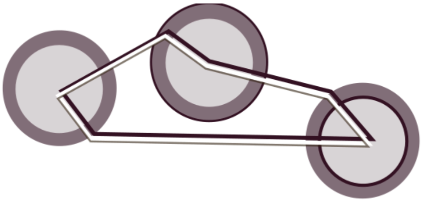
One node within an organisation to mention is the *team*. What is the definition of a team within the organisation? Is it tying a number of people working on the same project, client or area? Does the organisation have teams that feel too big or too small? How does the organization recruit team members?

There is a lot one can do to understand the health of an organisation's teams, but starting from data is always a safe move. Many companies nowadays make use of some work planning or organising software, and this software gathers a lot of very useful data. Taking a sample of teams which have varied in size over the years and looking at their Throughput (the rate at which a team outputs value generating items) and at their Cycle Time (the time it takes for the team to start and finish working on a value item), we can soon start to spot some trends.

Quoting *Klaus Leopold*, throughput "will increase because more employees can work on more orders", but cycle-time will also increase and thus the company's time-to-market. Why does it increase? In essence, it is because when an organisation recruits more people it recruits them to speed up the *active phases* (points in time where valuable work items are *actively* worked upon) of its workflow, when in reality workflow is composed of idle phases too (points in time where valuable work is *waiting* to be worked upon). The idle phases often occupy a bigger chunk than the active phases (Leopold estimates that these may take up to 90% of the entire workflow), and so the direct improvement attributed solely at scaling is minimal at best.

How would your organisation react should more people join its forces? How can you prevent impacting the Cycle Time into becoming so slow that drags the organisation into the abyss of irrelevance to the market?

Efficiency in the use of Energy and Resources



Behold, once again, for there is the verb “use” followed by the word “resources” in the same sentence, again a trigger for the curse of micro-management! Let us instead consider knowledge and the energy to transfer it for delivering value as the real topic we need to address.

When talking about organisms, for example, this topic is fascinating. Geoffrey West mentions that while there seems to be a “fixed number” of heart beats a mammal can have in its lifetime, small mammals (like mice) have very fast heart rates whereas large mammals (like elephants or whales) have really slow heart rates. The efficiency, therefore, is in the ability of the organism’s heart to distribute oxygen and other resources across the (space filling network of) veins and appendices of the organism. The size of the heart, in this case, has adjusted to make this transfer of energy and resources efficient.

Organizations too have their precious, vital resources to spread across its networks. These might be domain knowledge, alignment, decision-making and development of ideas, to mention a few, that will help the propagation of the business only if they reach all nodes and spaces. How does your organization cope with transferring knowledge? How does starting a new project or initiative feel in your organisation? Is it vibrant and responsive or too hectic and heavy? Do your efforts in building up enthusiasm fall on passive hearts?

The workflow (which we spoke about mapping in the first point) gives a clear indication of the flow of energy and its conversion into value for the organisation. The work items currently being worked upon can (should) be mapped onto the flow, and this will start exposing some patterns too, like bottlenecks, “invisible” waiting states and constraints. There may be a pattern of overcommitting inspired by the myth of multitasking. How would your work really flow if you consider setting up Work In Progress (WIP) limits? It is counter-intuitive in the beginning, but what this would imply is that you start shifting your organisation into thinking about finishing work rather than just starting it. When scaling up, more workers will pull more work, and with them bottlenecks and constraints. How can the organization prepare for this?

Conclusion

Scaling is often inevitable, but it can be postponed until the organisation is prepared for it, and this without missing any important opportunities.

In this article we have looked at the laws of scaling in nature and referred to principles and practices in Kanban to draw an understanding about an organisation’s readiness for scaling before it takes crucial steps that will either turn it into a dysfunctional organisation or else shape it into a strong value generating powerhouse.

First we looked at spaces the organisation needs to fill as a growing network, and we identified the market for tapped and untapped opportunities as an external space, and the organisation’s internal workflow as a space that organisations should look at to solidify their current capability of delivering value.

Then we discussed the teams as an example of an organisation’s nodes whose parameters will impact their ability to work effectively. An organisation should aim at having teams that have predictable throughput and cycle-time to be able to have a reliable and effective time-to-market.

Finally we looked at the organisation’s capability of distributing knowledge, ideas and its various (knowledge and creative) resources across its networks in a way that is efficient to its efforts.

We used Kanban to draw connections between the theory of scaling and practical approaches for systems thinking. In the coming articles we will dig deeper on how data can be used to decide the parameters of scaling and also on ways how organisations can grow leaner while maintaining, and increasing, their capability of delivering value.

About Matthew Croker

My journey has taken me through various roles: from care worker in children's homes to software development to Agile coaching. Consistent throughout my experience was the need for people to synchronize their energies, understand their environments, and work together in order to excel.

In the world of IT I have seen immense value in modelling and using historical data to give more light and colour to the decisions of the future. I have co-created the Decision Espresso as a tool for teams and organisations to structure their discussions towards fruitful and tangible action plans.

Ways to Contact Matthew

www.matthewcroker.com
matthewcroker@gmail.com
www.linkedin.com/in/matthewcroker

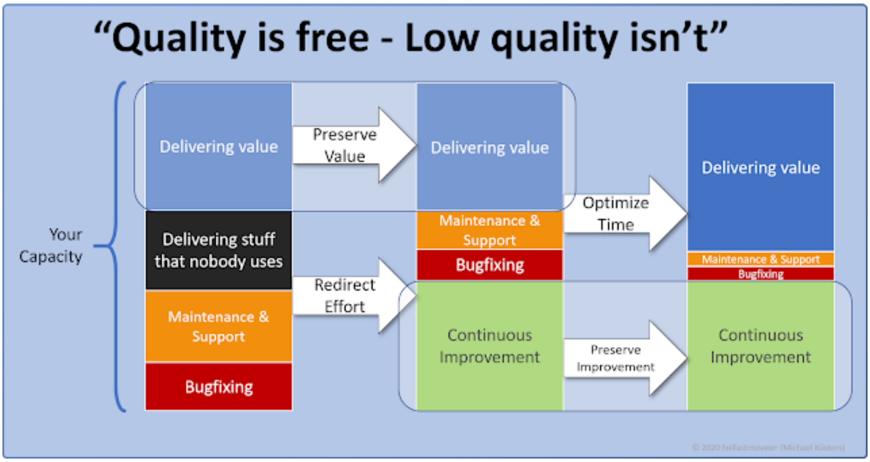
The Investment into Quality

By Michael Küsters

If you're setting out to "become Agile", or "more Agile", I would like to say something in words as simple as I can:

Unless you're willing to invest heavily into quality, forget about "Agile".

Now, what I mean with "invest in quality" is <u>not</u> "*throwing huge amounts of money into testing*", because the investment you will make is actually free, and it doesn't involve hiring additional staff, either: If you're doing it right, you will spend a lot less money to get better results. What you need to invest is attitude, thinking, brainpower, capacity:



Now, let me elaborate:

The commitment to quality

I think that no sane person would say that "*we like to produce garbage products.*" I have never met a developer who would state on their CV, "*I was producing crappy software.*" Likewise, I have never met a manager who would introduce their line of service as, "*We deliver crappy software.*"

If nobody would say that - then why is it even something to talk about?

Most commitments to quality are just lip service. You must act upon your commitment.

When I say, "commitment to quality", I mean that everyone, and that is, everyone, must continuously ask, "*How do my decisions and actions contribute towards quality, both in a positive and a negative sense?*" Any decision that leads to poor outcomes should be reverted, and any action that leads to poor outcomes should be stopped.

Nobody needs to justify themselves for *not* doing the wrong thing, or *not* doing something in the wrong way. It should go without saying that "*if you see that it's going to end badly, don't do it.*"

Managers must commit to creating an environment where team members have the freedom to do the right thing. Reciprocally, developers must commit to resolving any problems within their own sphere of control and naming any organizational barrier towards high quality, even if that barrier has been set up by the CEO in person. This must be unpartisan, free from personal preferences or fear (and these three things are already massive barriers to overcome!)

Everyone, and that means absolutely everyone involved, from managers over developers all the way to side stakeholders, must commit to do their best to enable high quality.

Quality thinking

You must turn a commitment into action, and for that, you must understand how to achieve quality. Quality isn't a local thing happening somewhere between a finished product and its users. Quality begins as early as in the ideation, and it never ends as long as the product exists. Quality is everything. It concerns everyone of us, and each of our actions.

Let's start simple, with the choice to add a certain feature to our product: Will it make the product better, or worse? For whom? How? In what way? How does the very decision of adding it affect both the process and the outcome?

The new feature could be a boon for some, and a put-off for others. It could make the product inconsistent with its purpose. It could make it clunky. It could put stress on development. It could have effects that are difficult to discover until it's too late. Could ... now I don't want you to over-think. At least, I want you to ask the questions that need to be asked, instead of simply shoving another piece of work into the pipeline: Oftentimes, the product could be better by *removing* a feature instead of adding another!

So, who has to think about the quality?

Designers, in design. Developers, as they develop. Testers, as they test (of course). Operators, as they operate. And that's the simple part: *"I must be mindful of quality in my work."*

Developers have to collaborate with designers. Testers have to do that, too. And Ops. Logically, testers also need to collaborate with developers and Ops. And everyone with the customer. And with management: "*We must be mindful of quality in our interactions.*"

All the time, everyone of us must constantly ask:

"What can I do, both in my own work, and to the work of others, to achieve higher quality outcomes?"

Quality practice

Move beyond words. Learn how to create quality: Design thinking. Behaviour-driven development. Test-Driven design. Clean Code. Stop the line. Data-driven decisions. Measure everything. Quality and Process management. Go-See. Standardization. Simplification. Candid feedback.

Just to name a few.

If you want to be agile, quality isn't just for testers: It concerns everyone, even the most junior developer and the most senior manager. Even those outside IT. While, obviously, I'm not going to ask a VP of Sales to write Clean Code, I would daresay that if *anyone* within the organization makes a choice that results in someone else breaking with an essential quality practice, this is going to hurt the company's bottom line. Hence, everyone must have a sufficient grasp on quality practices to maximize the overall outcomes of the company.

Everyone must understand enough about quality practices to do what it takes to get best results.

Capacity for Quality

I'm not going to sugar-coat this: If you don't have processes, infrastructure and code that are designed for quality, you're not going anywhere unless you put some effort into that.

You must allocate a certain amount of capacity for improving and preserving quality.

If your company has been spending $100k on shoddy stuff and could live with it, what's so wrong with spending the same $100k on something of higher quality? Nothing! Never accept shoddy outcomes: "*Do more*" has lower priority than "*do well.*" Capacity invested into low quality is *lost*. Capacity invested into high quality is *gained*.

"*But,*" - I hear you murmur - "*We will be slower, and the business / customer can't wait!*" - not really.

Stop thinking about what you deliver in a day or two. Set yourself a horizon of a month, half a year and then a year. Ponder how much effort you invest during that time into fixing bugs, into rework, into unused stuff and into rote work.

Set yourself a target to generate the same business results, without doing any of that wasted work. Then, ask yourself what you would need to get rid of this pointless work. If you manage to cut down on 10% of your current workload, you have gained 10% capacity for improvement. This capacity is now not "*free for* *more poor quality work*": It's free for *improvement*, so that you can do higher quality work that makes people more happy.

True Story.

Is all of the above a figment of imagination, wishful thinking, idealism?

No. I'm talking about measurable, tangible business outcomes.

Helping my clients engineer quality, we have significantly reduced defects, non-value adding activity and rework to *cut down new feature development lead times by over 50% and expenses by as much as 30%,* while reducing cost of failures on business end by seven-digit figures.

The "invest" we needed was about 50% of people's time initially - to change on their thinking, their practice and their environment, and 30% of their time subsequently.

If it was your money: Would you be willing to let people spend more of their time on quality, if the outcome was that *you have to spend less money to get more things faster, while earning more money* from happier customers, working with happier staff?

Yes, this question is purely rhetorical: "Time spent" is irrelevant if the outcome is "better business"!

Do it.

This article was originally published at
https://baa.tco.ac/3FCM

About Michael Küsters

Michael is an Enterprise Agile Coach, who supports everyone from garage startup to international corporations on their Agile Journey.

Michael has developed, tested, operated, architected and managed software systems at all scales and served as CTO in a medium-sized organization for more than a decade before opening his own company.

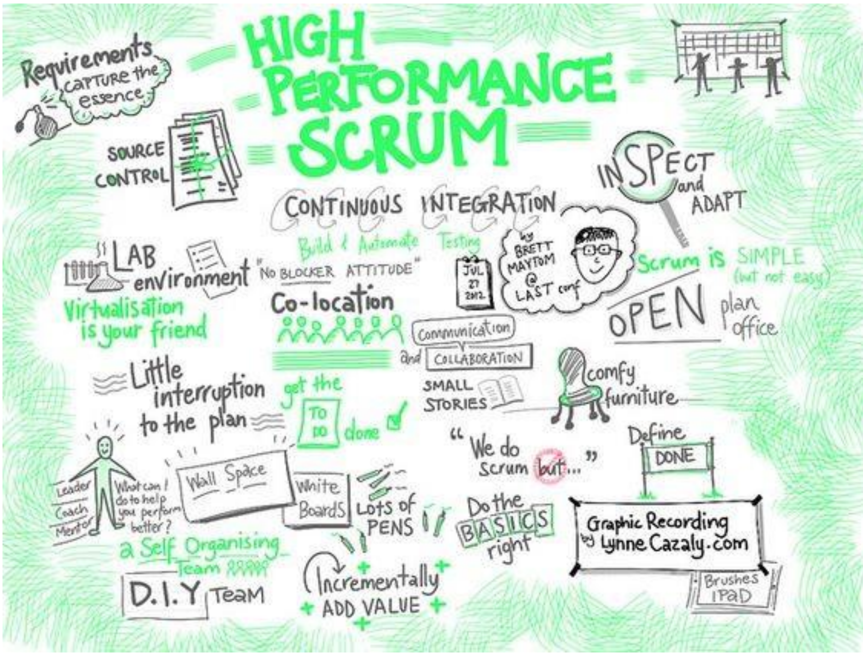
Drawing on this experience, he helps people learn and apply relevant techniques that help them to identify and systematically remove issues in their workplace, so they can deliver higher value and be more satisfied and successful in their work.

Ways to Contact Michael

failfastmoveon.blogspot.com
mkuesters1977@gmail.com
www.linkedin.com/in/michaelkuesters
twitter.com/failfastmoveon

Scrum Master's Journey in building high-performance teams.

By Muhammad Waqas Sharif



Scrum master is the guardian and protector of Scrum Team, someone that resolves impediment and has control over the scrum processes. Apart from the conflict resolution and facilitating ceremonies one of the most important responsibilities of the SM is to facilitate/coach/help the team in achieving the highest level of performance.

High-performance teams are not a myth, it is certainly possible as stated by Jeff Sutherland in his famous book “Scrum: The Art of Doing Twice the Work in Half the Time” that “a scrum team can achieve up to 400% of improvement in performance/productivity”.

Have you ever questioned yourself as a scrum master how is it possible? What are the common characteristics of such teams?

How can you facilitate a team to achieve such a level of success?

The high-performance team is one that always meets and repeatedly surpasses its goals. High-performance teams not only achieve excellent results, but they often do it uninterruptedly. The main features of high-performance teams need to be first known and then settled and sustain constantly during the team life cycle.

The first approach is to know about your team. You can do this by exploring/determining the maturity of certain characteristics in your team.

The other is to perform some quantitative analysis with your team so you can get a clear idea of where your team is standing right now. And what are the key areas where you need to invest your time more with the team to achieve the desired goal?

You can apply both approaches to the teams consistently producing results but they want to improve more and to those teams which are dysfunctional and you need them to be on track.

I will discuss both approaches in this article.

Characteristics of High-performance Teams

We've all experienced, or at least witnessed, high performing teams in action. We've also experienced, or witnessed, low performing teams (LPTs). What's the difference? What are their characteristics?

Dragana Hadzic explained these characteristics very well in her course *Building a High-performance Team*.

Team Purpose: The team should have a goal or a purpose to fulfill. Teams must know the reason why it was formed in the first place. It is more like a mission and for scrum teams, a product owner is a man who can explain this to the team. Each member of the high-performance team knows what to do? When to do it? And how to do it? There is a clarity of purpose and one that everyone is committed to.

Team Values: If we talk about scrum teams values come from the Scrum guide. Each member of the high-performance team should live the values. In low-performance teams, values are not shared as a whole and value for them are just the words on some poster.

Team Communication Manifesto: High-performance teams often set some rules for communication like no one will be using the mobile phones in the meeting or everyone should reach the meeting in time. For scrum teams, timeboxing rules come from the Scrum guide.

Psychological Safety: This is a part of the team's communication norms but I have listed it as a separate identity. Teams environment where everyone is able to speak up without fear of embarrassment or rejection. Many great ideas are lost otherwise. High-performance teams established such an environment so everyone can throw his/her idea or comment on anything.

Team Skills: High-performance team members need to learn new skills over time. Scrum emphasis on self-organized and cross-functional teams. Teams should have T shaped skills which means they develop additional and related skills keeping the end product in mind. They don't limit their individual contribution as per their existing skills.

For example, in a music band, the musicians should have an understanding of how multiple instruments are played to create the desired effect.

Modern Practices: High-performance teams always look for a better and quicker way to solve the problems. These teams are continuously inspecting and adapting their practices. Example of best practices is Test-driven development, Test automation, etc.

Process Adherence: High-performance teams adhere to the process and follow the guidelines by book. Like for example in Scrum, the team adheres to the timeboxing of ceremonies. And invests the right amount of time in the grooming the stories.

Adaptability: High-performance teams are agile and adaptable. They focus on the end result and are open to how that goal can be achieved. Low-performance teams are often rigid with rules.

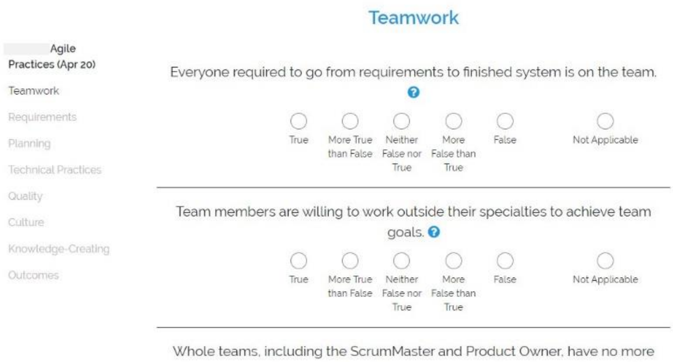
Characteristics Measurements:

After observing the above metrics you as an SM can list down the potential improvements for the team. You can use group or individual coaching to share the results and make a plan to work on each improvement in a prioritized manner.

Quantitative Analysis

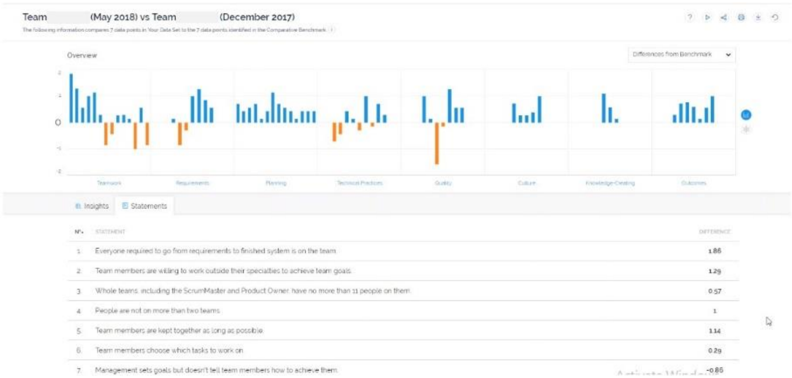
There are many agile assessment frameworks are available in the market which will help you to understand the current level of your team. There is a tool that I have used recently for the assessment of my team called Comparative Agility.

The tool is paid but a free version is available with some trimmed down features. The tool is based upon a survey that allows you to know how your team is working together. It leans towards Scrum so it is best suited for the teams working in Agile and Scrum framework. You can identify improvement areas for your team easily.



After you are done with the survey you can compare your levels with the industry benchmarks. This helps you understand e.g. how your team's technical practices are compared with those teams who are considered successful in the same domain.

This tool will also help your team in the long term by periodically repeating and comparing the previous and new results.



Conclusion

Scrum is based upon the empirical process control theory. Continues inspection and adaption will help your team grow to the level where the team always meets and repeatedly surpasses its goals. The role of the Scrum master is very important here to bring all the data and findings in front of the team and make a plan to work for the improvements. High-performance teams bring many benefits for everyone from companies to the individuals working in the team but the key to success is knowing yourself, your team and continuous improvement.

Credits

https://medium.com/swlh/how-to-facilitate-a-team-purpose-workshop-e90ddf536424

https://www.linkedin.com/pulse/8-characteristics-high-performing-teams-dr-tim-baker/

https://www.forbes.com/sites/forbescoachescouncil/2016/10/14/13-characteristics-of-a-high-performing-team-and-how-leaders-can-foster-them/#9e8bf7f394a8

*https://www.agilealliance.org/glossary/scrum-master/#q=~(infinite~false~filters~(postType~(~'page~'post~'aa_book~'aa_event_session~'aa_experience_report~'aa_glossary~'aa_research_paper~'aa_video)~tags~(~'scrum*20master))~searchTerm~'~sort~false~sortDirection~'asc~page~1)*

https://www.yodiz.com/blog/scrum-master-job-descriptions-and-responsibilities-in-agile-methodology/

https://www.pinterest.com/pin/445504588111299090/

This article was originally published at
https://baa.tco.ac/3El6

Kickstart a New Scrum Team

By Muhammad Waqas Sharif

In this article, I would like to share my experience on how I facilitated a kick-start/kick-off session for a team that is assembled for a new project at DPL.

According to Richard Hackman (Leading Teams) and Ruth Wageman (Senior Leadership Teams), a team's effectiveness is based on the way you launch the Team by 30%.

30% refers to how you start up the Team: define the Vision, Product Backlog, team, how Team handles the conflicts, clarify the process, coordination techniques, roles & responsibilities, etc. It takes 2–3 days to do this activity properly and has a high ROI.

Kick-start is important for any team therefore all the activities under the kick-start must be researched and prepared well enough to have the right impact on the team.

By keeping in mind, the importance and the impact of Kick-start, I started to prepare a week before the event. I spent a lot of time researching and finding out how best I can deliver the session? What should be the key ingredients? What activities should be included? How much do teams need to know about agile/scrum before they "go"? What do they need to know about each other and bout the product to be built?

A scrum master takes on the job of the teacher during the kick-start. There is much for the team to learn. If kick-start is done well, it can be jet fuel to a team, helping them to go further and faster than they ever imagined.

The major challenge for me was to conduct the session online as everyone on my team is working from home. I have to be extra careful about ensuring the participation of everyone and to make sure the interest and motivation of team members never drop during the session. To make sure everything goes smoothly I approached my session as follows.

- Video chat to ensure everyone's focus.
- Screen sharing for the presentation.
- Use of online real-time collaboration tool (Miro) for engagement.
- Microsoft teams whiteboard for clarifying concepts.

I divided the session into three parts.

- Getting to know your team.
- Scrum training.
- Creating a team working agreement.

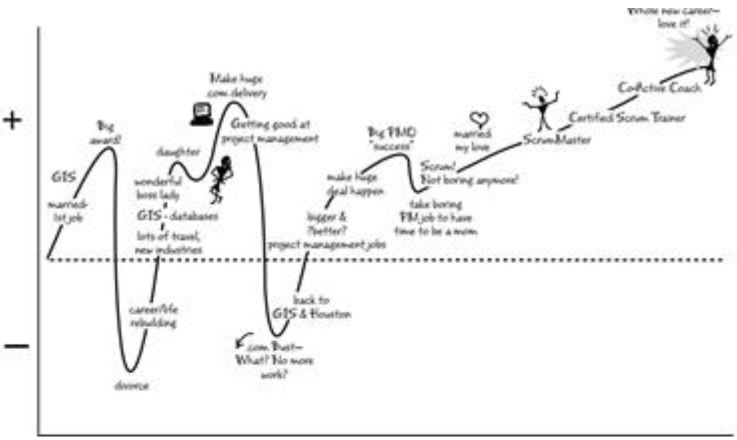
Getting to know your team

You can find dozens of excellent activities to help people break the ice and to get out of their shyness and comfort zone so they can share their dreams, skills, perspectives, and goals with one another. It usually doesn't matter if people worked with each other for a long time, or they are entirely new to each other. What matters is to spend some time during the first session as a team to get to know each other.

I have borrowed an excellent activity from the Lyssa Adkins book "Coaching Agile Teams". Activity is called Journey Lines.

"A journey line graphs a person's professional journey. Starting as far back as they want, each person draws the ups and downs of their journey on a flip chart. As you can see in the figure, the journey line often looks like a roller coaster with notes at each high and low point to remind the person of that particular event. Some include details of their home life as well as their professional life. Some stick to the professional facts. It doesn't matter. No two are ever the same, and they all work fine. Each team member will present their own journey in front of others.

Through sharing one's journey and being truly received by the other team members in the form of the notes, each person is affirmed for who they are and what they bring. And the entire team builds a foundation for working cross-functionally because they now know each other's backgrounds."



Journey Map from Coaching Agile Teams by Lyssa Adkins

I requested my team a couple of days before the activity to create their journey maps. And that they can draw it in the form of a map or just make a bullet list as per their own convince.

Few things that can be shared in this activity:

- Introduction
- Education
- Total Experience
- Technology stack/Key skills
- Previous projects
- Major accomplishments/Downfalls (overall and at

DPL)

- Goals for this project
- Long term Goals/New skill to acquire
- Anything you want to share with the team.

Quoting directly from Lyssa Adkins book to shed some light on the importance of this activity.

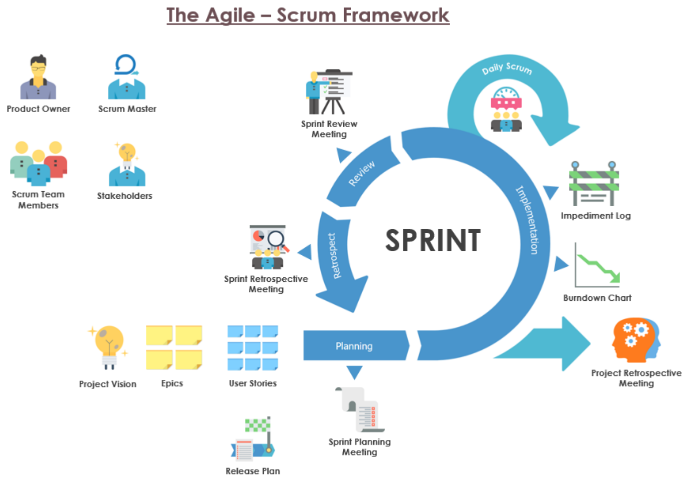
"The results of Journey Lines are sometimes astounding. One such case happened for me during a "new" team start-up. The team members had worked together for years over cubicle walls and had come together for their first-ever agile experience. As one person presented her journey line, she got to a hard bit — the year she had cancer. As she talked about the impact of cancer on her life, I noticed another team member getting emotional. After a few minutes, this team member started to cry. Through tears, she said, "I sat in the cubicle next to you that whole year, and we never knew about each other. I had cancer then, too."

Overall this activity brings a lot to the team. Even if deep connections don't appear to happen at all, the Journey Lines exercise plants the seeds by making the skills, expertise, and overall background of each team member visible to everyone.

Journey Map of one of my team member

As we are following scrum at DPL so it is important that every team member has some basic understanding of the Scrum framework.

As a Scrum Master, you might need to adjust your teaching style depending on the experience of your team. In case your team members have been working on scrums teams before, then your responsibility as a teacher is to make sure everyone on the team is on the same page as far as scrum knowledge is concerned.



Scrum Framework

There are chances each member has his/her own definition of the scrum but it's important to help and take them to textbook scrum. You have to show the original version of the scrum. And most importantly you have to do it in a polite way so no one feels offended.

You can start the session by saying something like this, "I know you all have worked in scrum projects before, but let's take few minutes so I can show you original version of the scrum to make sure we all are synchronized in our understanding of it."

If your team is new to scrum, you have the opportunity of a fresh start. Your session will be more detailed and longer as well. Try to cover every aspect of the scrum including roles, rules, events, and artifacts.

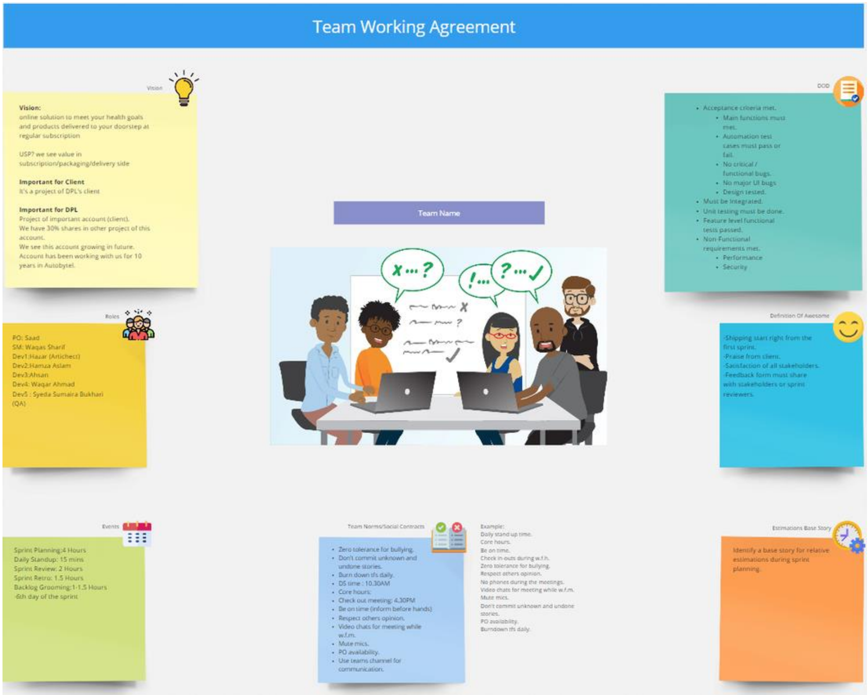
I covered Vision, Release planning, Estimations along with the roles, rules, events, and artifacts in my session.

Creating a team working agreement

You can call it Working Agreement, Team Norms, Team Contract, Team Ground Rules, or Team charters anything you want to call it.

Working Agreements are a simple yet powerful way of creating rules/guidelines for the team's culture. It is a list of standard procedures which is a valuable tool for guiding team conduct. It helps the team in creating an environment that is safe for everyone. Where no one is afraid of bringing up any idea to discuss with the team. These agreements usually enable the ethically correct environment and set the team expectation with each other.

I conducted this activity using (Miro) an online real-time collaboration tool to make sure everyone was engaged in the activity.



Screenshot of working agreement environment in Miro

I included the following things in our working agreement:

- Team name
- Vision
- Team roles
- Events
- Team norms
- DOD
- Definition of awesome
- Baseline story

Team Name

Name is important as it gives an identity. Every team must have a unique name that should represent rules, values, and principles on how people on the team want to work together. A name provides a bonding and a common purpose to fulfill.

Jacque Rowden, the Senior Director of Continuum's Help

Desk, told a story about a group that worked during the overnight shift. For a while, the group members would blame each other for any shortcomings that occurred during their shift. However, after they came together and rallied under a common team name, they were able to better allocate their resources and as a result, their shift began running much more smoothly and efficiently.

The name should be different than the project name and having a logo/flag will enhance the impact. It can be placed on the walls in team premises.

Vision

Everyone on the team must be clear about the vision of the product they are working on.

The Following questions provide the opportunity to better understand the overall vision.

- What is the core vision statement?
- What value does it provide to the client?
- Why it is important for our company?
- What problem does it solve for the end-user?

Team Roles

Identify all the roles on the scrum team. Make sure the team size is within limits as mentioned in the scrum guide.

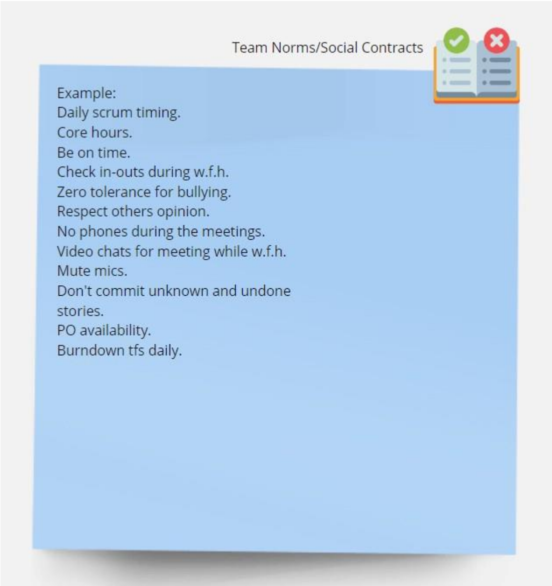
Write the name of members like who will be product owner, who will be scrum master, and who's part of the development team.

You must also list the core competencies/skills of the development team members.

Events

Specify when and where events will take place. Also, mention the event duration as well by keeping in mind the sprint length. The team must know who is required for which event. Don't forget to specify the time of backlog grooming by the team's coordination e.g "we will schedule it on the 6th day of the sprint".

Team norms



In this section, the team can lay down the rules they need to follow during the product life cycle.

A good way to make sure everyone remembers what they have included in team norms by displaying it on the wall in team premises.

Teams don't have a police force to ensure that it's followed. It's the responsibility of the team as a whole to make sure everyone follows it and respectfully reminders to peers when trespassing occurs.

You can include things like following in the team norms:

- Daily scrum timing.
- Core hours.
- Be on time.
- Check in-outs during w.f.h.
- Zero tolerance for bullying.
- Respect other's opinions.
- No phones during the meetings.
- Video chats for meeting while w.f.m.
- Mute mics.
- Don't commit unknown and undone stories.
- PO availability.
- Burndown is the responsibility of the team and tfs should be updated daily.

It is worth considering *Scrum Values* as a source of guidance while creating this list e.g. Value of Respect fulfills when everyone makes sure to be on time in meetings.

Definition of done

There must be a common understanding of what DONE means. A feature is considered as done when it is developed, tested, and meets all the acceptance criteria. Done also means that it is possibly (Product Owner decision) shipped. A great definition of done is the one the Team can actually commit to and not wishful thinking.

Sample DOD:

- Acceptance criteria to be met.
- The main functions must functional.
- Automation test cases must pass.
- No critical/functional bugs.
- No major UI bugs
- Design tested.
- Must be Integrated.
- Unit testing must be done.
- Feature level functional tests must be passed.
- Non-Functional requirements must be met.

Definition of awesome

After the done comes the awesomeness. I have borrowed this concept from an article by Faye Tsampardouka. I urge the team to move one step ahead and have some higher goals to hit. Teams need to answer the question: What will make them super excited and Awesome.

Sample definition:

- Be able to build, test, and ship a feature within a sprint.
- QA starts the very first day of the sprint.
- The work in progress is limited.
- The satisfaction of all stakeholders.
- Praise from the client.

Baseline story

The team brainstorms to identify a user story that will be used as a reference during the estimations. A baseline story must be picked with a common understanding and requires the whole team. After picking a story analyze it and estimate it. It's better to use story points estimations.

Most teams usually pick the smallest story for the baseline e.g. "A simple user login functionality". But you should facilitate and discuss and let the team decide what to pick.

Baseline story should be available all the time especially at the time of planning and it's a good habit to just revise the story during the estimations in each sprint planning.

How to maintain the working agreement

Working agreements should act as a living document. The document should be reviewed periodically as the team evolves. It is good to review it in every sprint retrospective meeting and see if teams need to add/remove anything depends upon the circumstances.

Ending

An effective kick start meeting serves the team a great deal. It helps creates a shared understanding, norms, and priorities for the team. A good kick start will help the team to achieve fantastic results a lot earlier than any other team does. Kick-start is important for any team therefore it's your responsibility as a facilitator that all the activities under the kick-start must be researched and prepared well enough to have the right impact on the team. This is based upon my experience and a lot of research I did to find the best activities out there on the internet. I hope this article will help all scrum masters to kickstart their teams and avoid a lot of problems scrum teams face during project development.

Credits:

- Coaching Agile Teams by Lyssa Adkins
- *https://www.continuum.net/blog/teamwork-makes-the-dream-work-the-benefits-of-having-employees-identify-with-a-team#:~:text=Encourage%20employees%20to%20come%20up%20with%20team%20names&text=Having%20a%20team%20name%20that,that%20is%20working%20closely%20together.*
- *https://www.arrkgroup.com/thought-leadership/work-*

agreements-for-a-scrum-team/

- *https://medium.com/@yaztokdemir/the-truth-about-working-agreements-5026c3952c73*
- *https://medium.com/the-liberators/how-to-kickstart-a-great-scrum-team-10-practical-things-to-do-2143bdde1a8d*
- *https://www.boost.co.nz/blog/2017/03/project-kick-off-kit*
- *https://tech.gsa.gov/guides/agile_team_working_agreement/*
- *https://build.thebeat.co/launch-event-how-to-give-a-new-scrum-team-the-kickstart-it-deserves-bb2eef9a7056*
- *https://agilepainrelief.com/blog/team-friction-inspires-working-agreements.html*
- *https://nomad8.com/articles/creating-working-agreements-that-are-actually-useful*
- *https://nomad8.com/articles/team-ground-rules-and-working-agreements*

About Muhammad Waqas Sharif

Muhammad Waqas Sharif is an MS-educated PSM (I – II) certified Scrum Master with extensive experience in facilitating teams to deliver proven results.

Being an agile explorer, servant leader, and facilitator, Waqas is adept at identifying impediments and problem areas and facilitating his teams in the collaborative implementation of corrective actions.

His mission is to guide, coach, and train companies and teams in their agile journey towards continuous improvement.

He believes that any team can be a high-performing team by:

-Living the Scrum values.

-Continuous improvement by Inspection and Adaptation

Ways to Contact Muhammad

medium.com/@m.waqassharif86

m.waqassharif86@gmail.com

www.linkedin.com/in/muhammadwaqassharif

3 Key Tips for Scrum Masters to Act with More Resilience

By Nagesh Sharma

How many of us believe that we are not operating to our fullest potential and there is always something we could do better, faster, and effective?

This blog is inspired by two books I read recently "The Power of Full Engagement" & "A-Team of Leaders".

In this blog, you'd be exploring 3 key tips to sharpen your focus and become more resilient.

Myth 1: Time management will help us achieve our goals

Many Scrum Masters I speak with often say that they need to become better at time-management as they are not able to make most of their limited hours per day, especially during the COVID situation. This is the typical mistake most people do by considering "time-management" as a means to "peak performance". However, in order to give our best, i.e., the full engagement we need to shift our focus from managing our time to managing our energy.

Time alone won't help us achieve our goals, we also need energy. So, instead of optimizing schedules, we must optimize our available energy. We typically tend to focus on physical energy, however, there are three additional sources of energy- emotional, mental, and spiritual that also require our equal attention.

- Emotional flexibility helps us to deal with pressure & stressful situations. For instance, when faced with a pressing business need, emotional flexibility helps us to stay curious and resilient to surmount the impediment
- Mental endurance helps us stay focused and keep us

engaged even in hard times

- Spiritual strength helps us stay connected to our deepest values that act as our true north

"Wherever focus goes energy flows" - Tony Robbins

Tip 1: Self Care is key to building your resilience. One effective way is to limit WIP, take frequent short breaks, and rejuvenate.

You may also experiment with Liberating Structures "*15% Solutions*".

Myth 2: Servant Leader makes the decision on behalf of the team

Most Scrum Masters perceive one of the servant leader stances as a decision-maker and fall into the trap. Great Scrum Masters design an environment in which the development team acts as leaders, self-actualize, and make their own decisions. The key is to find balance and accomplish shared leadership by designing the environment to give every team member a shared purpose.

To get a better sense of this, consider asking this question to a team member- What are you doing today? Do you want your team members to answer " I am solving a customer problem-for example improving the website performance" or " I'm just fixing some defects"?

When teams don't feel connected to a higher purpose or believe their work has meaning, they start channelizing energy away from work. The most successful teams I have come across had one thing in common - "cohesion" and the leader's job is to create a psychologically safe environment that supports cohesion.

There are two types of cohesion:-

1. Social Cohesion, which is about healthy relations and social bonds between team-mates

2. Work Cohesion, which is about bonding through a shared

commitment to achieving a common goal
Both are important to boost team morale.

"The stronger the WHY the easier the HOW becomes."

- Jim Rohn

Tip 2: Facilitate a team values co-creation exercise and connect everyone with a sense of purpose. Coach team on having solid Sprint Goals.

You may also experiment with Liberating Structures "*Purpose to Practice*".

Myth 3: You need to work hard giving up your joy to be successful

Joy serves as the basis for the company's culture. Joyful teams are more productive and engaged, leading to better results.

I love a great story from the book "Joy Inc."

Consider the difference between the wright brothers and Samuel Pierpont Langley. While both were striving to create the first-ever airplane, their motivations were different. Langley wanted fame and fortune; the Wright brothers wanted to experience the joy of flight. And it was the pursuit of joy that inspired the Wright brothers to soar.

"If your actions inspire others to dream more, learn more, do more and become more, you are a leader." - Simon Sinek

Tip: Design a Joy Driven culture in your teams and organization. Have shared habits and spend time together remotely by virtual happy hours or virtual water coolers.

You may also experiment with Liberating Structures "*1-2-4-all*".

Summary: Change, uncertainty, and instability are inevitable as it's necessary for creative environments. Promoting a joyful culture based on openness and transparency drives people to give their best. I'd like to conclude by asking this question. Before going to bed tonight take a few minutes and ask yourself "What are you paying attention to? Are you truly listening or listening to confirm your own beliefs?".

Scrum Master's Toolkit to Coach the Person, Not the Problem

By Nagesh Sharma

One of the key stances of the Scrum Master is being a Coach. Often coaching is perceived as the art of asking powerful questions. How many times while taking a coaching stance do you ever feel like you're chasing the coachee? Have you ever felt that the coaching sessions are going around in circles, and your coachee keeps resisting the main issue?

As a part of my journey of Accelerating Coach Excellence, I came across Marcia Reynolds. She is an amazing Coach and has written many books. After reading her books I got so inspired that I committed myself to share my learnings with the larger community.

In this blog, you will explore 3 common myths of coaching and how you can switch gears as a Scrum Master to get on track with your coaching conversations.

Myth 1: Coaching is synonymous to asking powerful questions

Have you ever observed a coaching session, what do you remember the most?

Observers often highlight the best questions asked. They don't recall the reflective statements that prompted self-reflection. The powerful question gets the glory. Coaching as a series of questions can sometimes feel like an interrogation, damaging trust and rapport.

Coaching is a reflective inquiry not a series of questions. In 1910, Dewey defined the practice of reflective inquiry in his classic book, "How We Think". The intent of inquiry is not finding solutions, but to provoke critical thinking about our own thoughts. Inquiry helps coachee evaluate their beliefs and clarify fears and desires affecting their choices. Statements that prompt us to look inside our brains are reflective.

Reflection includes:-

- recapping
- labeling
- using metaphors
- identifying key or conflicting points and
- recognizing emotional shifts

Inquiry combines questions with reflective statements. Questions seek answers; inquiry provokes insight. When you are racking your brain trying to find a good question, you are in your somewhere else but not present. Being present is more important than being able to ask a perfect question. You need to create a safe space where their creative brains are activated, safe enough to explore their own thinking, and take action. People are naturally creative, resourceful, and whole, this concept was first seen in the work of psychologist Alfred Adler.

As Daniel Kahneman says in his book " Thinking, Fast and Slow", we resist self-exploration especially when emotions are involved. We don't change well on our own. To stop adverse thinking patterns, someone outside our head needs to disrupt our thinking by reflecting our thoughts back to us and asking questions that prompt us to wonder why we think the way we do.

Tip: In your next coaching conversation, instead of asking questions like "what have you tried so far or what else questions", first reflect words and expressions and then ask a question with curiosity, not memory. You may try mirroring, summarize what you hear, don't hesitate to share inconsistencies or contradictions. Label an experience for clarification like "I noticed..." or "I heard ..." or "I sense ..."

"Be the powerful mirror for your coachee".

MYTH 2: Coaching is about asking Only Open-ended questions, not Closed.

Open questions that start with what, where, when, how, and who will get more than one-word responses. Open, exploratory questions incite a deeper look at what is prompting behavior or inaction. One-word answers to questions shut down instead of open up a conversation. These moments can make the coach uncomfortable, feeling there is nowhere to go in the conversation.

The purpose of questions is to disrupt a pattern or flow of thinking and prompt deeper exploration. The focus of a question should be on whether it opens or closes the client's mind. As long as the question furthers the conversation, it shouldn't matter how it is structured.

Closed questions are effective in at least three situations:

- To help clarify what coachee want to resolve in the coaching
- To affirm if a reflective statement is accurate
- To prompt coachee when it is clear they have had a startling insight but they aren't speaking

Closed questions that follow summaries, such as, "Is this correct?" or "Do you want to change this pattern?" or "Do you know if your expectation is realistic?" can help crystalize thoughts if asked with sincere interest.

Closed questions can be used to test the validity of a reflective statement. For example, when you summarize what is heard and expressed, notice shifts in emotions, or identify underlying beliefs or assumptions, you might ask if the coachee agrees with the summary, observation, or inference.

Tip: You need to create a bond of trust that deepens over time. This trust-based relation is key for being a coach and an effective thinking partner. When the coachee trusts you are there to help them achieve something important to them, they will accept the discomfort of an edgy closed question.

For example, if they realize their actions have been sabotaging their desires, you might ask, “Will you ever be content with the situation as it is?” or “Will you regret not taking action a year from now?” You might follow up on these closed questions with an open question about what they want to look at or do next.

Give up knowing, Be Curious. Mastery is the deepening of presence not the perfection of skills.

MYTH 3: It is easier to give advice than to take the time to coach others to find their own solutions.

Are you providing suggestions or advice taking an outside-in approach or invoking your coachee's potential by taking an inside out approach? Is your coaching transactional or transformative?

When you tell people what to do, you tap into their cognitive brain, If what you suggest relates to or affirms their current knowledge, they are likely to agree with you. They might have needed outside confirmation to fortify their confidence before acting. Offering ideas might sound like an efficient way to guide people’s actions. This is true, however, you also run the risk of making them dependent on you for answers or approval before they take action. The results are often counter-productive and lead to lower levels of self-organization.

Transformational coaching works from the inside out. Reflective inquiry is a powerful way to create disruptions in thinking that lead to breakthrough transformation and change. When coachee attaches new meaning to themselves and the world around them, their capabilities, their limitations, and what they define as the right and wrong shift. The shifts cause changes in their choices and behavior.

Neuroscientist Michael Gazzaniga says we spend our days in automatic thought processing, rarely stopping to question the reasoning for our choices. Someone outside our brains is needed to break through these protective devices with a transformational approach.

Tip: Practice the art of non-interference by refraining from giving advice or suggestions. Don't try to be a healer, fixer, or expert in order to be the coach. Consider asking yourself what triggers a breakthrough or self-awareness?

Summary: Here is a small self-assessment for you to know if you are ready to Coach people.

Do not coach if you can’t do the following:

- *Let go of how you want the conversation to go.* You want coachee to resolve their problems, but you can’t be attached to how the conversation will progress or what the outcome will be. If you can’t detach, you will end up forcing the conversation in the direction you want it to go.
- *Believe in the Coachees’ ability to figure out what to do.* Do you have any judgments about your coachee that could get in the way? If you doubt their ability to find a way forward, then choose to mentor instead.
- *Feel hopeful, curious, and care.* If you are angry or disappointed with the coachee, they will react to your emotions more than your words. If you are afraid the conversations won’t go well, do what you can to let go your fear so you model what courage and optimism feel like.

Not all conversations can or should be coaching sessions. Figure out what people & teams need and then choose to coach or take appropriate stances like teaching or mentoring.

About Nagesh Sharma

Nagesh Sharma's mission drives him in increasing the joy and accomplishment Organizations feel by delivering value early leading to Customer Benefits and Business Outcomes. Nagesh is an Enterprise Agile Coach, PST @ Scrum.org, ACE Cert. Coach by David & David, Intelligent Leadership Master Cert. Coach by John Mattone and ROI Cert. Expert.

Nagesh is also a certified Facilitator by Management 3.0 & Collaboration Superpowers. He is an active speaker at various international conferences like Scrum Day Europe, Scrum Deutschland, Scrum Day India, Agility Today, Agile NCR and more.

Ways to Contact Nagesh

flowsphereacademy.in/
nagesh.sharma@flowsphere.in
www.linkedin.com/in/nageshsharma08/
twitter.com/just4nagesh

Key Mindset Changes for the New Remote Manager

By Pilar Orti

Having always pictured yourself with your team around you, seeing them every day, checking that they're okay, overhearing conversations that reassure you that everything is in order, it must have come as a shock to find yourself completely disconnected from team members overnight - knowing that they are dealing with new challenges too, as you all pull together to go up a very steep learning curve.

And while the last few months have been interesting (ok, I mean challenging) for knowledge workers, they've also brought new challenges for those of us already working remotely. This kind of remote work is not what anyone was used to, and it's not what anyone wished for.

Not only were people unable to go to their offices or other chosen place of work, they were also unable to do all the other in-person activities that gave structure and meaning to their days. That's why, incidentally, it's important not to take the levels of innovation, productivity and creativity during these months as indicative of innovation and creativity in remote work under better circumstances.

Now that everyone knows how to use video meeting software (although some have still been struggling with poor internet connections), and we all understand that we need to start experimenting with asynchronous communication to make remote teamwork sustainable, there's a dawning realisation that the next stage is not going to be as simple as "everyone going back to the office" anytime soon.

It's time for managers to step up, and go through another major mindset shift.

The Shift To Remote Leadership

Shifting your mindset and shifting your perception of what a manager or team leader needs to be in the online space is critical.

So, in danger of simplifying what will indeed be a detailed, thought-through practice, these are five mindset changes that will help new managers of remote teams - or should I say managers of those teams new to working remotely - stay healthy and sane.

1. Adopt A Coaching Mindset

While we're looking at difficult shifts in mindset, adopting a coaching mindset will also help remote teamwork be more sustainable. This might involve:

- Taking the time to have a conversation with someone to think through their next actions (other than telling them what those actions should be).

Taking the time to ask questions that will help a person think through a problem and design their own solutions, rather than tell them how you did it when you came across a similar problem some time ago.

Granted, a coaching mindset might not have been what was needed over the last few months, when some people just wanted to solve the problems quickly and get on with the more difficult issues faced with at home. But now might be a good time to identify those simple opportunities where a question that prompts thinking will be more efficient than providing a solution.

As well as empowering your team and building their individual capabilities, this shift from being the fixer to the facilitator will also provide reassurance, and grounding in the belief that we are not fighting fires anymore, but transitioning sustainably into a new way of working together.

(For more on adopting a coaching mindset in a remote team, *read this past blog post*.)

2. Embrace Delegation, Avoid Interference

One of the issues with having teams *adopt Visible Teamwork practices* is that suddenly a lot of the work and a lot of the progress becomes visible to everyone, including the manager. Maybe even more visible than if we were sitting in an office together.

The challenge for the manager who likes to know what's going on every minute, every day; who finds comfort in knowing who's doing what and when, is that now they have all this information available to them. (The beauty of Visible Teamwork of course is that it's also available to all team members.)

Learning to seek information only when it's needed (as opposed to checking that everyone is doing their work), is a key skill for the new manager to learn - or else, your role can become exhausting, and your team members can feel self-conscious and micromanaged.

Furthermore, when you adopt the principle of "Open Conversations", it will be challenging to see team members discuss their work, and only interfere if we absolutely think they've gone down the wrong track.

Better still, you could choose not to read every single conversation and only reply when tagged. So long as your colleagues know they can get your attention in this way when needed, that might be much more efficient anyway.

(For more on this, read *The Dangers of Working Out Loud*.)

3. Make Friends With Your Technology

Technology is your friend, not least because it can help you have better relationships with team members.

It doesn't matter that you are a "people person", it doesn't matter if you've always seen technology as something that gets in the way of personal connection. The truth is that many remote workers have developed really strong relationships with others without ever meeting them in person and technology mediates every element of their collaboration and communication successfully.

But in order to use technology to help you build relationships, you need to learn to make it work for you.

Make sure to:

- Master your notifications (they are, after all, your friends). Choose what you want to be notified of, and choose what can wait until you're ready to go looking for it.
- Master your email client (if you must). Can you schedule your messages, so that you can get that email off your chest at any time, but it gets sent out during office hours? And make sure replies don't demand your attention at the wrong time.
- Learn to communicate clearly and succinctly by text. (I'm still learning that one!) Asynchronous communication gives everyone flexibility when communicating - but if they read your message four hours later, and they're still missing information or clarity, the conversation will draw out for days.

Before you send that message, think about the person receiving it: Have they got everything they need from me or are they going to have to come back for clarification or more questions? Conversely, while over-communication is the better side on which to err, check that you have not confused your intended message with waffle or padding.

4. Get Intentional

Working in a team in the online space requires formalising the informal. This is difficult to assimilate when you are someone who thrives on the ad-hoc conversations that emerge in the colocated workspace.

Spontaneous interactions will need to be planned. Habits and practices which emerge organically in the colocated workspace, now require deliberation. For example, if you want your team members to check in with each other regularly, but not to have to drop everything else to have a meeting at 9 am every day, you are going to need processes to communicate context, availability, and how are you doing in general. And you can't rely on real-time communication for this all the time.

Learning experiences and conversations that sometimes happen as we walk across the office will need to be given a space in your team communications. Informal learning can continue to be informal, but formalise where the conversations should happen, formalise a way of sharing learning within the team.

Do it simply, and in a way that team members can share their learning as it happens, or as they reflect on it. Open a shared document like a Google doc, have a Trello board just for learning, have a Slack channel, what app you use doesn't matter - but formalise how it will be shared, and give people the space to do it, in their own time.

5. Don't Make Yourself Indispensable

During the pandemic many managers have been the absolute linchpins that kept everyone together: the person that many have gone to for support, they are the ones that have made sure personal relationships in the team didn't break down; many have gone way beyond their call of duty for their team. Some managers have thrived, some of them have really enjoyed discovering deeper connections with team members - but others are on the edge of burning out.

If this last point resonates with you, it might be time to stop thinking of yourself as the team's lifesaver. You can't be the one continuously solving everyone's problems, the only person that team members come for support.

You need to start acting more as a connector, a facilitator enabling team members to help each other out and help your team members solve their own problems.

For those of us who really care, who really want to make a difference, this might be the most difficult mindset of all. But those of us who have been in the remote space for long enough, know that the beauty of using technology for teams to communicate is that self-organisation is much easier, now that we have various apps that allow us to share our work and its progress.

Now might be the time to think: “Is there anything I (we!) could do, so that team members find it easier to help and ask for help from each other? If I disappeared for a month, what would happen? Is there a simple process we can set up now, so there is a web of support, for team members to easily help each other out?”

What Next?

Regardless of where you are in your relationship with “remote” (whether you are looking at ways of making remote teamwork sustainable or looking at how to work as a hybrid team), I invite you to use the above points to help you plan the next few months. Even if some of you are gradually returning to the workplace, continuing to think of yourselves as a remote team can help avoid the divide between those who are in the office and those who aren’t.

What can you begin doing as your team’s leader, or what online communication practice can you adopt as a team?

More importantly, what behaviour, actions or practices should you stop doing, that are getting in the way of your team adapting to the online medium?

This article was originally published at
https://baa.tco.ac/3Ewj

About Pilar Orti

Pilar is the Director of Virtual not Distant, where she helps managers and their teams adopt online collaboration practices. She also helps organisations keep their dispersed people connected through internal, employee-led podcasts.

Pilar is a prolific podcaster herself, hosting the 21st Century Work Life podcast and others. She's also published a range of books. (And she's the voice of pilot Xuli in the BBC's animation GoJetters.)

Ways to Contact Pilar

www.virtualnotdistant.com
pilar@virtualnotdistant.com
www.linkedin.com/in/pilarorti
twitter.com/PilarOrti

Systems Thinking Approach to Implementing Kanban (STATIK) for Scrum Teams.

By Richard Dolman

STATIK (Systems Thinking Approach To Implementing Kanban[1 2]) can be a great technique to help teams get up and running quickly, even teams that are using Scrum.

Applying the STATIK approach leverages the first principle of the Kanban method, "Start with what you do, or know, now", in order to help avoid over-thinking how existing work, structures, and/or roles "fit" within Scrum. To be clear, I am not talking about what is commonly referred to as "Scrum-ban". Applying this approach designed for Kanban can dramatically accelerate the forming and improvements of a Scrum team.

As a long time practitioner and coach of Scrum, I've observed many teams struggle getting started with the framework. Change is hard for most people and we often get resistance at first. Even though the level of change is relatively low with Scrum, it can be a challenge for some teams to get started in certain environments.

A couple of anti-patterns I see repeatedly include:

- Management struggling with how to address new role definitions like Scrum Master and Product Owner among other aspects of the framework. This causes anxiety and distractions for teams who simply want to get started working together.
- Teams jumping into Scrum without getting grounded on the nature and characteristics of their work or ecosystem. This often means the teams are just going through the motions and may be slow to build shared ownership of their work and outcomes.

Over the years I've tried a few different approaches to help teams get started. I learned about STATIK from a colleague several years ago and began applying it as a model for my Team

Chartering workshops. I have used this approach to help teams form and establish norms for them to work toward–including how they will norm around Scrum.

The STATIK method helps answer the following questions:

- What is the purpose of the team?
- What are the products or services we provide and to whom?
- What is the current way of working?
- What are the sources of dissatisfaction?
- What are the sources of demand?
- What are our current capabilities?
- How does work flow through the team/system?
- What kinds of services are there?
- How to design a kanban system that allows us to visualize and manage the work?

For the purposes of this topic, I will be referring to the "kanban system" (lower-case 'k') rather than the Kanban (upper-case 'K') method, since I'm not limiting the use of STATIK to teams implementing Kanban. The kanban system represents the elements and boundaries of the team's work. For a Scrum team, that may include from the point an item enters their backlog until the point they declare it "done".

Here's how it works –

At a high level, STATIK involves the following steps:

1. Understand 'fitness for purpose' and identify sources of dissatisfaction
2. Analyze demand
3. Analyze capability
4. Model the workflow
5. Discover classes of service
6. Design kanban systems
7. Roll out – socialize and negotiate, measure, inspect and adapt…

Using STATIK with a Scrum Team

Setting up the workshop: To make it relatable, I provide a "Story" card for each of the steps and set up a simple kanban board to visualize the activities as we progress through the workshop. For each Story, there are Acceptance Criteria that represent the activities that we will work through together and enable the team to agree on what is 'good enough' for the team to move forward.

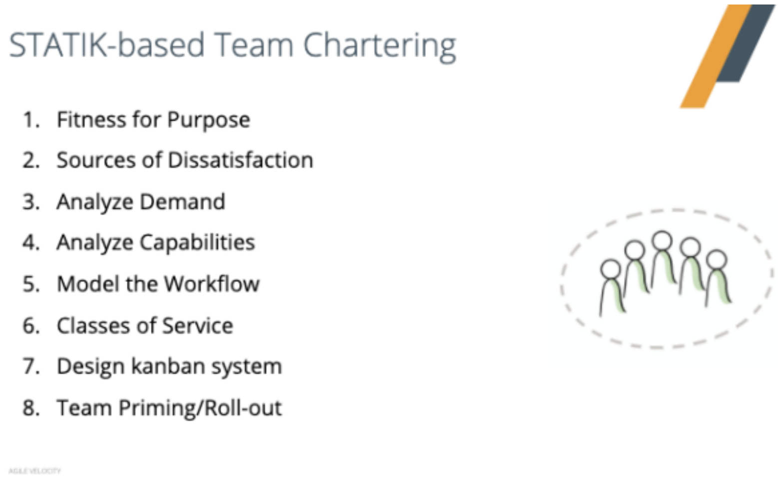


Image1 - Sample STATIK Workshop

This is how I design the chartering agenda

As with any team activity or workshop, it's important to properly set the stage and get everyone connected. A simple connection exercise could be to ask the participants to consider and discuss the following:

- What are the major characteristics of our Work, our Team, or our Process?
- How do these characteristics impact our delivery?
- How do these characteristics influence our decision-making?

This gets them thinking beyond just "We do 'X'" and opens up the door for the Systems Thinking aspect of this. *NOTE: Depending on the knowledge and experience of the team, you may need to provide a brief overview of Systems Thinking[3]. I've generally found it's not a critical barrier to running this workshop, but a little overview never hurts.

Step 1: Understand 'fitness for purpose' and identify sources of dissatisfaction

The intent of this step is to explore the criteria that define customer satisfaction and define the "fitness criteria" such as lead time or quality. These determine how the customer evaluates whether the service or product delivery is acceptable, or "fit for purpose."

When working with teams that are applying Scrum, I will adapt this step (since Scrum addresses these differently than Kanban) by discussing fitness criteria, as well as conditions for success and the norms that they will need to agree on in order to fulfill their purpose. The team will create their Definition of Done in steps 5 and 6 and establish appropriate mechanisms to measure and enable customer satisfaction.

This step can be facilitated as a single story, but I tend to break it into two separate stories so that the team has a full appreciation of their current state before moving on to the next steps.

Story 1: As team members, we want to define and validate our Purpose and our ability to fulfill that purpose, including conditions of success, so that we have a strong sense of our team identity.

Acceptance Criteria

- We have drafted a mission statement and team slogan
- Our Values and supporting Norms are known (We believe in… Therefore we will…)
- We have defined high-level Conditions for Success

- We are able to serve the needs of the Customer

Story 2: As team members, we want to discover areas for improvement, what our customers and stakeholders are saying, and delivery or quality challenges we are facing, so that we can be more customer-focused and start building a backlog for continuous improvement.

Acceptance Criteria

- We have exposed all relevant Sources of Dissatisfaction and/or things we struggle with
- We have visibility to and understanding of Customer and/or Stakeholder feedback
- A backlog for Improvement has been created or updated

When forming *new* teams, Story 1 is a great way to get teams started and centered on their team identity. Prompting questions can be something like–"Why are we being formed as a team?", "What do we stand for", "How do we want the rest of the organization to view us?".

At this first level, I try to emphasize fun and safety, keeping it light and even a little idealistic. Things get much more detailed and analytical later, so providing engaging and uplifting conversations right off the bat sets a positive tone. During this step, I typically encourage brainstorming, but will also work in some form of diverge-converge activity to get ideas generated and shared.

A few things that quickly emerge here–introverts & extroverts, visual thinkers & critical thinkers, as well as those who "just want to start doing the work" & those who "want to feel like we're a real team". Don't try to solve or fix any of that. It is, in part, what will define their norms as a team.

When working with *existing* teams, Story 2 may be a bit more impactful. Hopefully, there is data/feedback available from a prior Retro and directly from customers and stakeholders to leverage. If so, we will pull in the relevant data and/or feedback and go through some sort of a "reconnection" activity, asking everyone to write a few stickies (physical or virtual) for each item from the past Retro, relating to how it impacts them or their thinking. We may do some dot-voting on items to see what people view as most important or impactful. If there is no prior feedback or data to reference, this presents an opportunity to do a Retro and seek feedback from others.

Before deciding what format would best serve the team, I probe into their norms and experiences. For example, if they're not doing regular Retros–why is that? Or, maybe they've been doing them, but they don't seem valuable or meaningful to everyone.

From here, I can then choose the approach that would be most impactful, rather than just repeating a technique they've seen before. One technique I find works well, in this case, is using the Sailboat/Speedboat format[4]. This is a great visual, metaphorical way of exploring how well, and under what conditions, a team is performing. We can glean insights and specific, practical ideas that will be used on upcoming steps.

If there is no recent customer or stakeholder feedback available, we will craft a simple survey or questionnaire that can be sent out. In this case, keep it simple, but try to create a sense of urgency so the team can get feedback quickly that they can use right away.

Step 2: Analyze demand

Story: As team members, we need to analyze sources of Demand, so that we have a shared understanding of who is requesting services/product from us and the characteristics of the items in our backlog.

Acceptance Criteria

- ID primary (if not all) sources of demand – including from whom/where, frequency, complexity, and impact of requests
- Understand the various characteristics of the requests and how that translates into work in progress
- Agreement on critical gaps or risks to our ability to effectively deliver

Here we explore Demand. What are the characteristics and origins of our work? For a Scrum team, some may simply say, "we pull work from the backlog that our Product Owner has prioritized". While this may be technically correct, we need to go beyond that to understand all of the sources of demand, as well as answering questions like:

- How frequently does demand come to us? (known as 'arrival rate' in Kanban)
- How complex is the work?
- How much variability do we have?
- What are our customers asking for?

I find it valuable to start with Stakeholder mapping. There are a myriad of formats, but I tend to start with a basic context diagram to visualize the team's key stakeholders. Are they consumers or contributors, internal or external? Understanding these first two dimensions can help answer questions about how we communicate or interact, how frequently we interact, and where they fit in our overall workflow.

Once we understand Demand, we can then analyze the team's capacity, or capabilities, to serve that demand.

Step 3: Analyze capacity

Story: As team members, we need to analyze sources of Demand and our Capacity to respond, including the skills and capabilities we need, so that we have confidence in our ability to deliver effectively and optimize our work.

Image 2 - Market of Skills Flipchart (drawn by a colleague)

Acceptance Criteria

- ID primary (if not all) sources of demand – including from whom/where, frequency, complexity and impact of requests
- ID our Capacity to focus on the work at hand, including the core skills and capabilities needed to deliver. (Initial

Skills Matrix developed)

- Agreement on critical gaps or risks to our ability to effectively deliver

This is another step that I will adapt for Scrum teams. I will use one of two techniques: a) Market of Skills activity or b) Skills Matrix creation. My choice of technique depends on how I interpret the teams' level of creativity or analytical preference.

The Market of Skills approach tends to suit teams that are more playful, open to exploration, or may have more of a "Clan/Collaborative" or "Adhocracy/Creative" cultural orientation.

Market of Skills is gamification that uses the metaphor of a market where you may sell goods. In this exercise, you are selling your skills and unique capabilities to your team members. When using this technique we may include a variety of skills beyond just the technical skills needed for the job. Someone may include musical skills or that they're good at puzzles–this makes it fun and helps team members connect with each other on a personal level.

The Skills Matrix modeling may fit better with teams that are "all business", very serious about their work and/or more of a Hierarchy/Controlling or Market/Competing cultural orientation.

A Skills, or Competency, Matrix activity is more direct and analytical. Step 1 is focused on identifying the technical and soft skills needed to do the work. Step 2 is to have each team member self-assess their competency level with each skill. Teams may choose how they want to reflect this–using a numeric scale (0-5), shading a quartile (see example), using symbols, etc. The key here is to encourage safety and honesty without judgment.

After each team member has self-scored, the entire team can then review the matrix and discuss the patterns they see. Where are the gaps or weaknesses? The team can then discuss how they will deal with that.

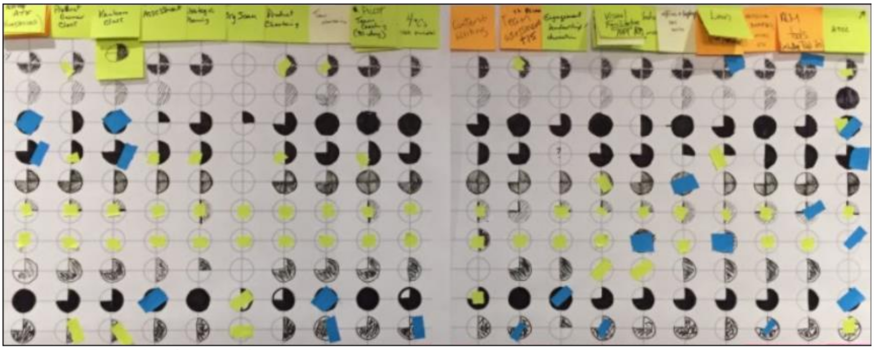


Image 3 - Example of a teams's competency map

The bonus benefit of this technique is that the team can also use the Skills Matrix to identify knowledge sharing and cross-training opportunities. If a team member is an “expert” in a specific area where others may be novices, they may agree to form mentor-mentee relationships.

While these techniques are most commonly associated with forming new teams, I have worked with many existing teams that have never done this sort of analysis and benefit from it as well.

Step 4: Model the workflow

Story: As team members, we need to model how work, information, and decisions flow through our team/system so that we have visibility to where we will need to define explicit policies and potential bottlenecks as we design our kanban system.

Acceptance Criteria

- Visual representation of how work flows through the team – May start with “most common” items and/or “happy path” workflow, then iterated to refine for exceptions or more rare scenarios
- Activities that provide the most knowledge are identified
- Tag potential bottlenecks or gaps that will need to be

resolved

I often find this activity to be the most interesting and impactful, particularly for existing teams. When I ask "Does everyone know what our process is for getting things done?" everyone usually nods and says "yes, of course." However, once they start visualizing it, it's not uncommon to see multiple versions emerge. It's not that people don't know their process or how work and decisions flow through the team. But, until they start mapping it out, they hold a bunch of assumptions and interpretations that may or may not be valid. That's the power of collaboratively modeling the workflow.

Image 4 - Facilitating a team workflow mapping activity (pic taken by my colleague Erika Lenz)

David Anderson[1] recommends asking the question, "What activity gives us the most knowledge?" I love this question. It helps the team think beyond the tasks they perform and gets into the meat of their learning and decision making. Building "why" and "how" into the system is just as important to the "what". I encourage teams to map this into their workflow/kanban system and to incorporate it into future retrospectives.

Depending on the size of the team, we may break out into multiple, small groups (diverge), then re-group to compare and contrast. Ultimately, the whole team needs to agree on a reasonable representation of how their work, knowledge, and decisions flow through their system. (converge).

For new teams, it's obviously a little different. It can vary a bit depending on whether they are: a) newly formed, but experienced with the context or delivery approach, or b) new to everything – each other, the work, the context, etc. In either case, it's more of a greenfield discovery activity and is generally done as a whole team rather than the diverge-convert approach. For new teams especially, it's important to remind them that "perfection is the enemy of progress", meaning we're not going to get it perfect the first time. We just need to get it "good enough" so that we can start to do the work, then inspect and adapt as we go.

Image 5 - example workflow mapping artifact

In all cases, it's not uncommon that the team will identify multiple "what if…" or "yeah but…" scenarios. If time allows, explore those that may be critical or impactful. For the rest, we tag those as "exceptions" and may add an item to the backlog to iterate through those in the near future.

Step 5: Discover classes of service

Story: As team members, we need to define the various classes of service that will require specific, Explicit Policies (rules), so that we are able to design our Kanban system to effectively optimize flow and give everyone awareness of the ”rules of engagement.”

Acceptance Criteria

- Key Classes of Service are defined, including prioritization, service level agreements, definition of done, etc.
- Team agrees on all explicit policies (rules) related to each Class of Service

A Class of Service is a set of policies which describe how something should be treated.

These definitions provide clarity on how to prioritize items as well as what and how we may handle an item, how to decide which items are more urgent–needing to be worked immediately–versus items that can wait.

A common example for a Scrum team might be to define how they handle new development work versus product support or maintenance work.

Often, the sources of dissatisfaction activity can uncover new or different Classes of Service. Teams may also need to define items with different business risk profiles that need different policies. Class of Service definitions (Explicit Policies in general) help teams deal with potential challenges from stakeholders (e.g HIPPO or WSFL challenges), as well as issues related to flow, WIP limits, and bottlenecks.

Another common definition for Scrum teams is the Definition of Done. This is a form of an Explicit Policy and may be defined as a general definition for the team or it may vary depending on the rules or boundaries associated with a particular Class of Service.

Step 6: Design kanban system

This step is one of the reasons why I think STATIK is so effective. We don't jump into designing the kanban system until we've done the discovery and analysis in the earlier steps.

Story: As team members, having now defined and analyzed the primary nature and conditions of our work, we need to design our kanban system, so that we can begin managing and improving our way of working

Acceptance Criteria

- All Team members agree to an initial kanban system that will enable them to begin managing and improving the flow of work
- The kanban system is visible*

Scrum teams often take the value of a well-designed kanban system for granted. After all, the Sprint is set and the Sprint Backlog is our "To-Do", so all we need is to show "Doing" and "Done". Right? Well, not always. Scrum was designed to deal with complex, adaptive problems.

So, our kanban system should be a proper representation of the complexity and adaptiveness that the team needs to manage. Many teams have both new development and production support to deal with, multiple products that they're responsible for, or multiple external dependencies that cause wait states or other flow variances. The more complexity that a team has to deal with often means they need a more sophisticated kanban system.

There is no right or wrong way to design your kanban system. I encourage teams to be creative and emphasize the things that enable flow and collaboration.

Step 7: Roll out–measure, inspect, adapt…

To close out this workshop, we do a "team priming" activity. This activity was co-create with a former colleague of mine, Daniel Lynn. He and I have paired on countless workshops and engagements.

Story: As a team, we need to be able to start doing and measuring our work and find out what our current impediments are so that we can hit the ground running and ideally be able to deploy an increment or change to production in our first sprint.

Acceptance Criteria

- Create a flipchart or other artifact reflecting:
- Tools we need
- Knowledge we need
- Environments we need
- People we need
- Identify and communicate impediments to delivering a shippable product increment
- Create the team's Technical Backlog (user stories w/ acceptance criteria)
- Communication plan/feedback loop for stakeholders

In this step, we discuss whether the team will maintain both a physical and electronic kanban board. Most, if not all, of the electronic tools have both Kanban and Scrum views built-in. I encourage teams to maintain both a physical and electronic board (often within a tool like Jira or VersionOne) for a while to give them both the tactile and electronic experiences in parallel. It's much easier to experiment and update a physical kanban board than to change a template in a tool.

However, given the global remote workforce trend (new normal), having a single physical board is not as feasible as it once was when we had more co-located teams. So, instead I recommend using a virtual "whiteboard" tool, like Mural or Miro, or something simple like Trello, that can be easily modified and experimented with in lieu of a physical board.

Roll it out, socialize it, experiment, and… inspect and adapt.

Closing

Whether you're launching a new team or already working as a team with Scrum or any other approach, consider trying STATIK to see if you can gain better awareness, alignment, and visibility into how the team works.

What ideas or concerns has this article raised for you? If you've tried and/or adapted STATIK with your team share your experiences and insights below.

Additional Resources

- David J. Anderson – STATIK article *– https://www.linkedin.com/pulse/statik-systems-thinking-approach-implementing-kanban-david-anderson/*
- Mike Burrows – Kanban from the Inside *– https://www.amazon.com/Kanban-Inside-Understand-connect-introduce/dp/0985305193*
- Peter Senge – Introduction to Systems Thinking *– https://youtu.be/eXdzKBWDraM*
- Luke Hohmann – Sailboat / Speedboat retro technique *– https://www.innovationgames.com/speed-boat/ https://www.tastycupcakes.org/2011/06/speed-boat/*
- David Hawks – Kanban vs. Scrum – How to Choose? *– https://agilevelocity.com/kanban-vs-scrum-how-to-choose/*

This article was originally published at
https://baa.tco.ac/3Evk

The "Agile C-Suite" and The Critical Role It Plays in an Organization's Path to Agility

By Richard Dolman

The recent Harvard Business Review (HBR) article "*The Agile C-Suite*"[1] and Forbes article "Agile Isn't New: What's New Is The C-Suite Embracing It" have prompted some good discussion around the Agile Velocity virtual water cooler.

While there is general agreement with most of what was published in these two articles, enough to elicit the ubiquitous 'thumbs-up' 👍, there's a fair level of "Yes, and…"

As a coach, having worked with countless organizations and executives over the years, from fortune 100 level to start-ups, I've reflected on all of the patterns I've seen displayed by C-suite level leaders and what differentiates those that have helped create sustainable agility versus those that don't.

With that in mind, I started to dig into these articles and relate to how we're addressing organizational leadership with *Path to Agility*®. This post focuses on the HBR article–I'll address the related Forbes article in a separate follow up post.

HBR article setup

Before digging in, it's worth noting that this article is based on an unnamed company and its CEO, 'Brian Johnson'. So, there are many clarifying questions I'd like to ask in order to have better context for evaluating the efficacy of the points being made, but I'll press on with full awareness that there are underlying assumptions that might be false.

Balance from the Top

I wholeheartedly agree with the need to "create balance from the top" as a central tenet of leadership engagement. However, the case being made in this article is that "Agile is primarily for innovation". This is common misrepresentation and limits the C-suite's thinking and ability to fully embrace agility. Agile is about being adaptive and creating the conditions (mindsets, capabilities, environments, etc.) that allow us to respond to change. If you include Lean thinking in the definition of "Agile", we can include seeking continuous improvement anywhere in the organization.

So, an alternate definition *for creating balance at the top might be to gain an understanding of complexity, by applying the Cynefin Framework*[2] *or a Volatility – Uncertainty – Complexity – Ambiguity (VUCA) filter, as the foundation for decision making to enable agility wherever it is needed.*

Balancing the Agile Enterprise

I appreciate the concept of the "business operating system" and balancing the agile enterprise. I like the visual tool and the overall concept acknowledging that there is a spectrum of "static" to "chaotic" for various "components" of the enterprise, along with an indicator of where an "ideal agile balance point" might be–where the organization is today and the balancing direction needed to close the gap.

This is, of course, all very subjective, but it may be a valuable tool to help tune decisions and prioritize areas of need.

The so-called "ideal" balancing point is not known, it's a hypothesis that needs to be tested and validated. It's also relative to what is known today versus what will be known in the future. So, there is a need to keep probing for where you are and where you think you're going.

The authors explain that in order to find the optimal balance, "the agile leadership team typically begins by creating new metrics to help determine how agile the company is, how agile it should be, whether it is moving in the right direction at the right speed, and which constraints are impeding progress."

I may just be splitting hairs here, but in my experience, we don't look to C-suite level leadership to create new metrics. Instead, we expect them to articulate clear objectives and a strategy that provides a mechanism to measure and communicate progress toward the transformation goals.

The first steps in the Path to Agility framework involve working with C-suite and senior leadership to define and align on their Compelling Purpose (why the organization should change). Aligning on and prioritizing critical Business Outcomes will guide the overall strategic vision for the Agile transformation.

Then, we typically see the creation of *meaningful metrics* and the assimilation of empirical data at the System and Team levels, with the front-line managers and teams that are closest to the work and the data. Top-down definition of metrics often comes with a ton of anti-patterns that can undercut the organization's ability to measure progress.

The Agile Leadership Team

In describing the Agile Leadership Team (ALT), the article states that, *"Typically, the agile leadership team includes part or all of a company's executive committee. At a minimum, it consists of the CEO and the heads of finance, human resources, technology,* *operations, and marketing—the individuals most critical to the components of the operating system."*

Yes, and there are often multiple conversations to be had in order to get the right mix of leaders and influencers on an ALT. For starters, whether or not it's feasible to include the CEO and all other C-suite level leaders likely depends on the organization's size and complexity. In most organizations, it can be very difficult to get C-level executives to commit to the level of engagement being suggested–although I do applaud the intent behind it and agree we should strive for it.

The formation of an Agile Leadership Team (ALT) is a central component to the Path to Agility framework. The ALT is a powerful guiding coalition that has the ability to drive change throughout the organization. This is founded on *John Kotter's 8-Step Process for Leading Change*[3] *– Step # 2: Build a Guiding Coalition* has been proven over and over as a central element to creating sustainable change. A common pattern we encourage when dealing with large scale organizational transformations is to have a Guiding Coalition (GC) for the enterprise transformation and an ALT within each portfolio group.

Even in smaller organizations, where multiple portfolios may not be an issue but C-suite bandwidth is still a challenge, forming both an ALT and a GC in parallel is an adaptation that I've seen be highly successful. The ALT is composed of key leaders, as high level as possible, but is not limited to the C-suite. ALT members may be at any level of leadership and have the ability to influence and drive change. However, there may be larger organization-wide impediments or changes that do need C-level ownership. The GC is made up of the most senior leaders that are needed to deal with the highest-level, most complex changes. The GC may not be a standing agile team, but are effectively on-call to deal with challenges that the ALT members need help resolving and are ultimately responsible for ensuring alignment of Business Outcomes and Strategic Vision.

Key Points worth Elaborating

"To create a truly agile enterprise, C-suite must embrace agile principles."

Yes, and one can argue that Agile is simply a set of values and principles, so by embracing Agile principles, you're "being" agile. When I'm working with leaders, one of the questions we begin with is "What is Agile?". My basic definition is "Agile is a set of values and principles, focused on teamwork, value for the customer, and adaptiveness, that provide a mindset for decision making." So, it is important to get grounded on principles.

Unfortunately, this has become a bit of a cliché. One can embrace a principle, without really having to change anything. In principle, I believe in exercise and a healthy diet. In practice, I may avoid working out and go for an extra scoop of ice cream. Leaders need to go beyond just embracing principles–they need to also model the behaviors and mindsets they expect from the rest of the organization.

"Rather than predict the unpredictable, agile leaders build rapid feedback loops."

Absolutely yes, and aside from setting and articulating the Vision and Purpose, the ALT's primary role is to nurture and amplify feedback loops. This helps build an organization-wide mindset around inspect & adapt and continuous learning. When done effectively, everyone starts to feel confident that they have a voice and are in the loop, which helps support a culture of shared, collective ownership.

The article properly points out *that "An agile environment has a way of challenging leaders"* with a certain style or mindset, like an old-school style 'command and control' leader. The article goes on to say, *"Agile, in short, requires humility from leaders."*

A resounding yes!, and this is where it gets messy… change is hard. Changing one's mindset, after years of success and validation that "Doing things my way is what got me into this position", is very long and non-linear evolution. Humility is a central characteristic of "post-heroic" leadership, which shows up as openness to change and a willingness to rethink basic assumptions.

The article concludes with – *"Agile teams often cite leadership and culture as the greatest barriers to the successful scaling of agile. But most leaders aren't fighting agile. They simply haven't understood how it applies to their roles or how to perform those roles in ways that enhance agility."*

Yes, and this is supported by the annual "State of Agile" survey conducted by Collabnet/VersionOne –

And while I agree that most may not be fighting agile, many are likely fighting change itself, overtly or not. It's a natural human response when faced with a threat to your norms, expertise, or status. Resistance to Change is ranked on-par with culture and leadership.

Bill Joiner's Leadership Agility[4] model suggests that only 10% of leaders have mastered the level of agility needed for consistent effectiveness. That's the post-heroic level mentioned above. The vast majority of leaders are stuck at the "Expert" level which limits their thinking and behavior to more tactical orientation along with a lot of "I/Me" focus. For these leaders the approaches and tactics suggested in this HBR article may be reasonable–give C-suite executives specific responsibilities and tactics in order to personally embrace agility. But the question remains whether that will be sustainable and yield significant change if leaders don't also evolve their leadership styles and fundamental ways of thinking to align with and support agility.

Summary

The journey toward organizational agility can be long and challenging. Our experience and research has shown that the vast majority of organizations who take on an Agile transformation will either experience “superficial agility” which usually results in resorting to old, ineffective behaviors, or see “pocket agility”, where some things may improve, but falls well short of the true organizational improvements needed to be more resilient. I think that is due, in large part, to the inability of leadership to personally embrace agility and at the same time create the conditions for the rest of the organization to do so.

We are able to address this with the Path to Agility, in a couple of ways.

We’ve designed the model in two primary dimensions:

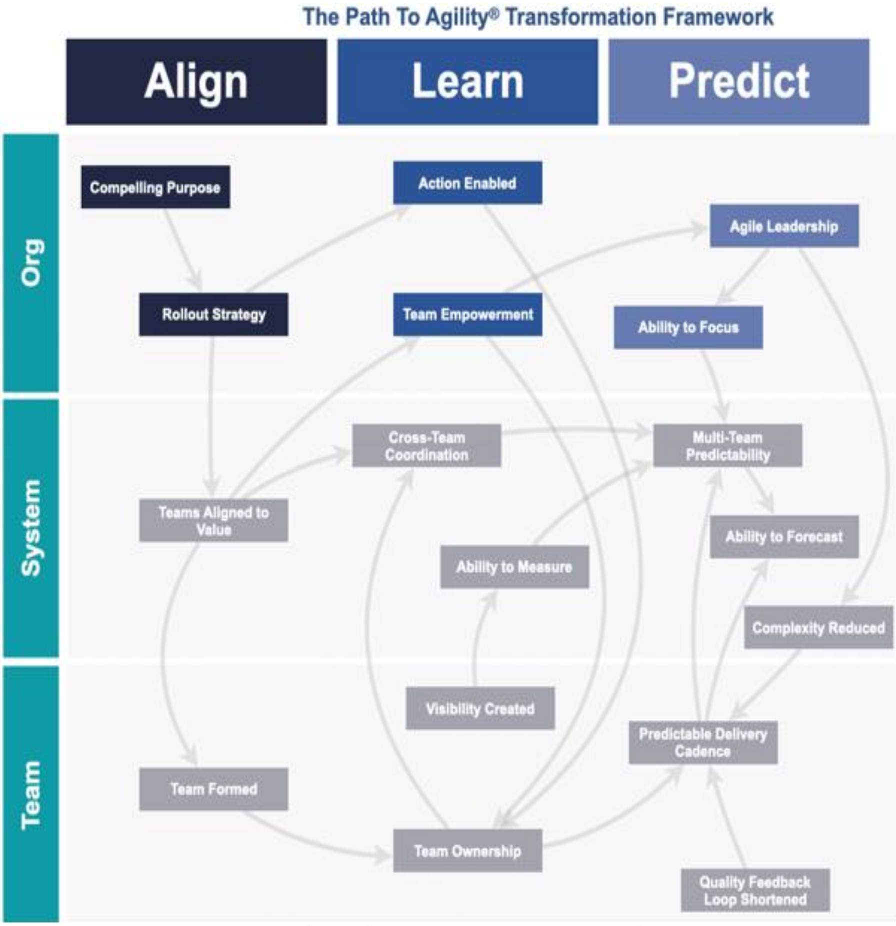
5 Stages of organizational change – Align, Learn, Predict, Accelerate, and Adapt.

In our experience, these are the typical stages that an organization needs to transition through in order to achieve a sustainable Agile transformation. Each stage provides a point of reference and context for change.

As I mentioned above, change is hard and it takes time, but there also needs to be a sense of urgency. Leaders need to establish a delicate balance of urgency and patience–understanding where you are along the path and knowing what and how to respond given your relative stage is crucial.

3 Levels for development and improvement of Agile Capabilities – Organization, System, and Team.

The *Organization level* is where Leadership develops a modern mindset, increases visibility throughout the organization, creates alignment around vision, goals, and measured success for sustainable organizational change. C-suite leadership is



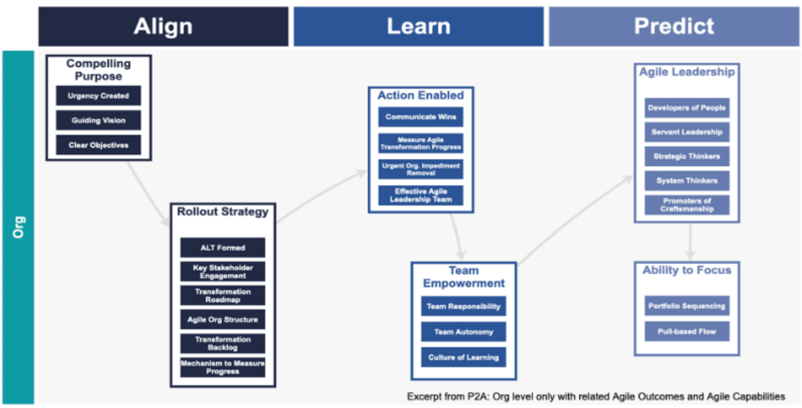
Excerpt from P2A: Levels and Stages (Align - Learn - Predict stages only)

ultimately responsible for ensuring outcomes and capabilities are being realized across the organization and by truly embracing an agile mindset and modeling those behaviors the rest of the organization will follow.

For each level and stage, there are a set of *Outcomes* and *Capabilities* that have been mapped and validated based on proven patterns through our collective experience leading Agile transformations. These components of the Path to Agility provide a framework and roadmap for leaders to improve transformation consistency, quality, and results.

We help leaders, C-Suite or otherwise, get started on their path by establishing an understanding of the advantages of an outcomes-based approach and taking responsibility for them:

- *Compelling Purpose* – the reason(s) why the organization should change. Creating a sense of urgency, defining and communicating a guiding vision, and setting clear objectives for everyone.
- *Business Outcomes* – alignment and prioritization of the core need(s) for the business. This, by definition, means we're engaging executives from across the organization, not just within IT.
- *Rollout Strategy* – establishing an ALT and a transformation roadmap, taking into account new organizational structure demands, key risks, and incremental rollout. This also involves engaging key stakeholders and designing the mechanisms to measure progress, then empowering action at the System and Team levels.
-



Excerpt from P2A: Org level only with related Agile Outcomes and Agile Capabilities

This framework provides leaders and organizations with clarity and confidence as they take on the challenging work needed for organizational agility.

Closing Questions

If you're a C-suite or senior-level executive leader –

- What are the keys to your ability to internalize and model agility?
- What are the primary impediments to embracing or sustaining agility?
- How can you create the environment where change is enabled?
- Do you have a support system, a peer, a mentor, or a coach?

If you're a front-line Manager or Coach –

- How do you get C-suite level leadership to go beyond simply 'supporting' agility to actually being the embodiment of it for the whole organization to see and follow?
- What are some obvious impediments to this that we've seen and/or solved?
- If you put yourself in the shoes of a C-suite Executive, what ideas can you take action on?

Reach out if you'd like to discuss or debate any of the ideas or perspectives shared in this article.

Acknowledgments:

1. Darrell K. Rigby, Sarah Elk, and Steve Berez Bain & Co. – HBR article May-June 2020 (*https://hbr.org/2020/05/the-agile-c-suite)*
2. Dave Snowden – Cynefin Framework (*http://cognitive-edge.com/videos/cynefin-framework-introduction/*)
3. John Kotter – 8 Steps for Organizational Change

(*https://www.kotterinc.com/8-steps-process-for-leading-change/*)

4. Bill Joiner – Leadership Agility (*https://www.amazon.com/Leadership-Agility-Mastery-Anticipating-Initiating/dp/0787979139*)

About Richard Dolman

Pragmatic enterprise agility coach, trainer, mentor, practitioner, explorer and continuous learner I try to help people and organizations unlock their hidden potential.

As an Certified Enterprise Coach (CEC) and Certified Agile Leadership (CAL) Educator, I work with executives, managers and teams to guide them toward sustainable growth and continuous improvement. I'm fortunate to work with awesome people, who inspire and challenge me.

Ways to Contact Richard

info@agilevelocity.com

www.linkedin.com/in/richarddolman

twitter.com/RichardDolman

This Is Why Your Company Doesn't Make Real Progress, Although Everyone Works So Hard

By Sahin Guvenilir

Photograph by theceomagazine.com

Treacherous bottlenecks can nullify all your hard work, no matter how industrious you are.

Have you ever wondered where "bottleneck" term comes from? Let's try to understand it a bit better.

Imagine you have a bottle of water and your goal is to empty it as quick as possible.

Scenario 1: You overturn it, the water runs out by itself, and it's done.

However, you realise that the bottle neck (literally the narrower part of the bottle) limits the flow speed of the water.

It makes you wonder…

*Scenario 2***:** So you decide to get another bottle of water. Identical. You overturn it, but this time you give the bottle a few turns to generate a vortex. The waters run out almost two times faster than the previous bottle.

Why? Because, in the second scenario, you created a vortex which allows the air in so the water can run out through a circling motion. You let the water and the air have their own separate paths to achieve an uninterrupted and continuous flow. A better flow.

What is going on here? You are basically *improving the flow of the constraint (the bottleneck).* You are emptying the bottle with a better lead time by optimising the whole.

Takeaway: If you want to improve the whole system, first, you need to identify and improve the systemic constraints (any limiting factors) that stand in the way of achieving your goal. Otherwise, any improvement initiative will result in a local optimisation that does not provide significant benefits.

In software development context, it doesn't work differently.

Although it can be argued that figuring out the systemic constraints is not that straightforward, it actually works in a similar way in software development context.

Imagine you have a team that works on a specific feature. Their technical excellence is top-notch, they use the most advanced tools, they have a great team identity. They are capable to build DONE increment in short cycles. However, the organisation has a policy that allows the team release their increment only after a painful process that is governed by external approvers and only as a part of a monolith code base in sync with other teams, on a specific date.

Do they get any benefit from having a high performing team, as long as there's a constraint in the process that prevents them from getting an early feedback from the users/stakeholders?

Does the organisation use the potential of this team?

What does this delayed releasing do to team morale?

Does that process help anyone or does it just sub-optimise the whole?

Let's make it clear, it does not necessarily mean the team would not consult anyone outside of the team to deploy their code to the target environment. There might be intentional 'organisational conventions' to comply with some legal requirements, or due to nature of the business.

Basically, there may be a context behind this scenario. That's more about being able to ask: "Is this process leading to a systemic constraint? Can we improve it?".

Now, there are two dimensions.

1. Do you think the culture of the organisation allows people to ask these questions without feeling uncomfortable?
2. Are the employees professional and courageous enough to speak truth to power?

My two cents, it's never only one of these, both are important. Top-down and bottom-up.

5 frequently encountered systemic constraints.

In the last 10 years, I worked in different industries including e-commerce, fintech, advertising and real-estate. Regardless of the field, I observed a pattern around the frequently encountered systemic constraints.

Although these constraints are mostly fluid and interconnected, I can mention *5 categories* and some powerful questions under each to initiate useful conversations.

Constraint 1: Heroic Management & Leadership Approach

- Are the people with politic influence (leaders, managers, or whatever you want to call them) acknowledged as heroic experts or as servant leaders who encourage survivable experiments?
- Is the manager role defined by the head-counts reporting to them or by their capability to create space for the growth of their people and teams?
- What kind of policies are in place to incentivise servant

leadership?

- Do the managers do *Gemba Walk*? Or are they astronaut managers who operate outside the atmosphere and can't see what's really happening on the ground ?

About Gemba Walk:

'Don't look with your eyes, look with your feet… people who only look at the numbers are the worst of all.'

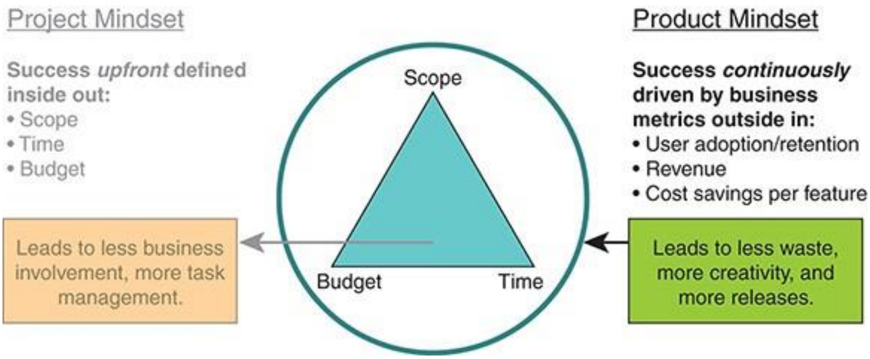
Taiichi Ohno — The father of the Toyota Production System

Constraint 2: Fortune-telling Driven Development & Budgeting

- Do the people know the difference between *'forecast' and 'commitment'*?
- Who decides 'how much can be done' in a specific time-line? Is it the people who do the work or someone else?
- Do you operate with *long-term budgets*? Does it create a pressure to trade off talent quality with speed in hiring to make you look scale quick?
- Does your Product Owner/Manager know how much a release costs?

Constraint 3: Ambiguous Product Definition & Measuring What Doesn't Matter

- Are you a project or product company? Do you define success inside-out or outside-in?



The Professional Product Owner By Don McGreal and Ralph Jocham

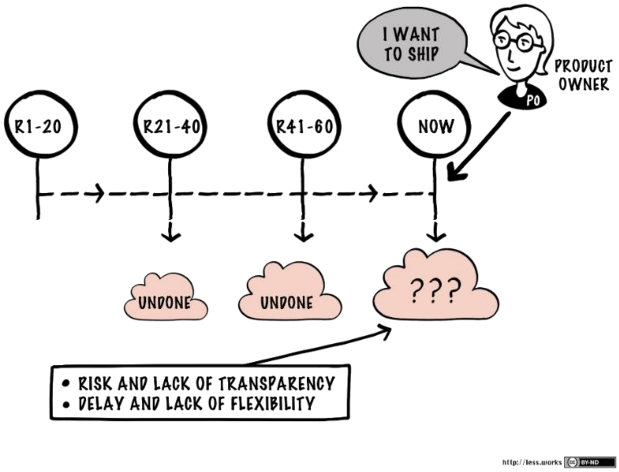
Even if you are a B2B company, delivering on time, budget, scope does not necessarily mean you deliver something that will make your client happy. These are circumstantial metrics and you will find more value in identifying better indicators that address outcomes rather than outputs.

- Have you defined 'what value means' in your context? Have you determined what your leading & lagging indicators are, in conjunction with the people who set the direction and the people delivering the value?

Some resources I've found helpful for perspectives in different levels and needs: *The 2019 Accelerate State of DevOps*, *Evidence Based Management*, *Actionable Agile Metrics for Predictability*

- If you hand out a pen and a paper to random people in your company, can they write the same thing about what your product (or service) is and why your customers use it? Not only business people. Everyone.
- How broad do you define your product? How do you create the right balance between cognitive capacity overload and broader learning opportunities?
- Do you have someone ultimately accountable to maximise the value of the product? If multiple teams are working on one product, do you have THE product backlog that is ordered by this person?

Constraint 4: Lack of Shared Understanding around DoD



https://less.works/less/framework/definition-of-done.html#UndoneWork-RiskandDelay

- When you mark something as DONE, can you immediately release it or is there a delta?

- How does this UNDONE work impact the teams in terms of planning capability and context switching? Do they often complain about allocating capacity for the bugs coming from an environment beyond their Definition of Done?
- Do you do anything to shift left and close this gap?

Constraint 5: Teams Structure (managing dependencies over eliminating dependencies)

- Is the current structure allowing the teams to own end-to-end flow of the feature/service/stream they are responsible for?

- Do the teams complain about spending too much time to manage the dependencies they have with other teams?
- Does the team have enough space to focus more on quality, innovation and to improve technical excellence?

Team Topologies introduces "Enabling Team" concept to close the aforementioned gap and support stream-aligned teams through technical consultancy.

These "enabling teams" actively avoid becoming "ivory towers" of knowledge that dictate technical choices for other teams to follow. They operate with servant leadership approach.

Conclusion

All in all, any progress other than in the bottleneck is an illusion. It is vital to discover and be aware of your systematic constraints that impede the flow.

There is no silver bullet. There is no one size fits all. Nonetheless, In my experience, if you can achieve an organisational mind-shift as below, you will improve the 5 aforementioned constraints.

- Servant Leadership over Heroic Leadership

- More Adaptive Planning over Big Budget Up-Front
- Product Mindset over Project Mindset (Outcomes over Outputs)
- Delivering a Small DONE Increment Earlier over Delivering Everything Later
- Eliminating Team Dependencies over Teams Managing Dependencies

As a final point, one should not forget that the organisational structure can also have an influence both on the company culture and the design of the software created (*Demystifying Conway's Law*).

This article was originally published at
https://baa.tco.ac/3FD1

About Sahin Guvenilir

Sahin is a seasoned Scrum Master, Coach and Professional Scrum Trainer from London, UK.

Throughout his career, Sahin has helped many organisations in various fields, including but not limited to FinTech, E-Commerce, Real Estate and Energy. He has supported more than 100 teams to deliver highest value possible sooner and manage their risk exposure continuously.

Ways to Contact Sahin

www.scrum.org/sahin-guvenilir

sahin@vaxmo.co.uk

www.linkedin.com/in/sahinguvenilir

Understanding Technical Debt with an Example

By Steve Matthew

Today, I thought I'd explain the concept of Technical Debt, a concept I personally struggled to understand for quite sometime. I used to think that a defect was Technical Debt because I often heard people saying that Technical Debt is a shortcut or a quick and dirty solution. As a result my mind kept associating that to bad quality code and defects because, how could something like that ever meet the Definition of "Done" and be of releasable quality?

After doing a lot of research, asking questions and getting clarifications, what I learnt was that you could create an Increment of releasable quality, that meets the Definition of "Done" and yet incur Technical Debt. Confused? So was I, and I am hoping that through a small code based example I can help make sense of this concept.

So what exactly is this shortcut or the quick and dirty solution that people keep talking about?

Consider the below C# code which takes a random integer between -5 and 5 and then displays the output as "negative" or "non-negative".

```
//Example 1:
```

```
// The very first time I wrote code looked something
like this.

int input = new Random().Next(-5, 5);
string classify;
if (input >= 0)
{
classify = "non-negative";
}
else
{
classify = "negative";
}
Console.WriteLine(classify);

========== Output ==========
// When input = 2
non-negative
// When input = -4
negative
```

Now, consider the below example.

```
// Example 2:
// After several months of experience and feedback, this
is how I would've wrote the same code.

int intInput = new Random().Next(-5, 5);
string strClassify;
if (intInput >= 0)
  {
    strClassify = "non-negative";
  }
else
  {
    strClassify = "negative";
  }
Console.WriteLine(strClassify);

========== Output ==========
// When input = 2
non-negative
// When input = -4
negative
```

Notice the differences?

Is the readability much better with the use of indentation? Yes.

Can the type of the variable be easily determined irrespective of where in the code base it may be used or re-used? Yes.

Is the output still the same? Yes.

Now, consider the final example,

```
// Example 3:

int intInput = new Random().Next(-5, 5);
string strClassify;

strClassify = (intInput >= 0) ? "non-negative" :
"negative";
Console.WriteLine(strClassify);

========== Output ==========
// When input = 2
non-negative
// When input = -4
negative
```

Notice how the number of lines have been reduced and how the same if-else logic has been represented by a single line of code?

Notice how the output still remains the same across all three examples?

Examples 1 and 2 show how technical debt exists yet the code still works as expected i.e. it is in a "Done" state. Example 3 perhaps is the most efficient way to write the same piece of code, thus reducing/eliminating the Technical Debt because the efficiency is much better when the same functionality can be expressed in lesser lines of code.

This is a very small example of what Technical Debt looks like and should *not* be used as the *only* definition, example and occurrence of it. In this small example, the issues that arise with Technical Debt may not be that apparent as this is just a few lines of code. However, imagine a very large code base with thousands of lines of code and similar opportunities for improvement or making the code more efficient. That situation represents the state of the Technical Debt in your code base.

Examples 1 & 2 may be considered "quick and dirty" because it is very easy to quickly write it up, but Example 3 may require some thought, knowledge and experience. It is more polished and has a finesse to it. I've heard some expert developers referring to such code as the mark of software craftsmanship.

Have you ever heard your Development Team members complain how there is more work than what they originally estimated when they actually looked into the code base? or that they need to fix something before they can actually start writing new code? Most often, it is something like this that they were previously not aware of that causes their frustration, as the Technical Debt might have accumulated to a very great extent.

What we can summarize from this is the following:

1. Technical Debt is *not* a Defect. A defect will prevent an Increment from being "Done"
2. The functionality of the developed code does not change as a result of incurring Technical Debt as evident from the examples above i.e. "Done" work could still potentially have Technical Debt.
3. Technical Debt may be intentionally incurred to save time, perhaps to get code into Production faster. In such cases, this should be captured in the Product Backlog and repaid as early as possible. The Product Owner can make a decision after consulting with the Development Team on whether to incur this debt in the first place. Similarly, the Product Owner can make the decision to repay the debt if he/she feels it is economically feasible or necessary for the future stability of the Product.
4. When Technical Debt accumulates, it may become harder to scale and more effort may be needed to reconcile the debt which in turn may affect your velocity to take on new development work.
5. Technical Debt (in my opinion) is not the same as "Undone" work. "Undone" work results from work that was previously considered "Done" but now is longer

applicable as the Definition of "Done" gets updated due to experience, new knowledge or new standards.

6. It is best to avoid Technical Debt in the first place. Follow the Lean principle of Build Quality In.
7. Make Technical Debt transparent by adding it to the Product Backlog.
8. If you do incur Technical Debt, let us take a cue from Xtreme Programming and make it a practice to refactor, refactor and refactor!

I hope this was a good read and it helped you understand this concept a little better.

This article was originally published at
https://baa.tco.ac/3FD8

Which Scrum Event Is the Most Important?

By Steve Matthew

"Which Scrum event is the most important? Explain your choice?" is one of those typical interview questions I have experienced from potential peers or hiring managers.

Sometimes, you are even forced to choose one and explain. Sometimes, I hear people saying that the Sprint Retrospective is the most important event because that's when you get to identify improvements or that is when learning happens. I disagree with this absoluteness and I will be addressing this later in this article.

In my opinion, this is either a trick question or rather a weak one. As a trick question, this gives the interviewer an opportunity to truly assess the candidate's understanding of the Scrum events, however, if the interviewer, forces a candidate to choose from one of the Scrum events, then it immediately becomes a weak question and exposes the lack of depth and understanding of the interviewer.

I'd like to address this question and state that all Scrum events are important and it wouldn't be right to single out a particular Scrum event as more important than the other. Each event in Scrum serves a specific purpose and other than the Sprint every event in Scrum is an opportunity to inspect & adapt.

Other than the Sprint itself, which is a container for all other events, each event in Scrum is a formal opportunity to inspect and adapt something. These events are specifically designed to enable critical transparency and inspection. Failure to include any of these events results in reduced transparency and is a lost opportunity to inspect and adapt.

Pay attention to the above excerpt from the Scrum guide. It talks about how the Scrum events enable transparency and as a result of that transparency, provides an opportunity to inspect and adapt.

Scrum is founded on empirical process control theory, or empiricism. Empiricism asserts that knowledge comes from experience and making decisions based on what is known. Scrum employs an iterative, incremental approach to optimize predictability and control risk. Three pillars uphold every implementation of empirical process control: transparency, inspection, and adaptation.

Now, let us look at the above excerpt from the Scrum guide. In summary, it talks about how Scrum is founded on empiricism. It also talks about how transparency, inspection, and adaptation are the three pillars of empiricism or empirical process control.

If we tie the information from these two excerpts together, then the understanding is that the four Scrum events (Daily Scrum, Sprint Planning, Sprint Review & Sprint Retrospective) help enable empiricism.

If we look at the definition of empiricism from the above excerpt, then what we can interpret is that the knowledge we gain through experience and what is known can help us learn, improve and even adapt immediately. There is no need to wait.

Although improvements may be implemented at any time, the Sprint Retrospective provides a formal opportunity to focus on inspection and adaptation.

The above excerpt hopefully strengthens the points I am making so far and here I'd like to address why the Sprint Retrospective should not be singled out as the most important Scrum event.

Please don't get me wrong, I am not saying that the Sprint Retrospective isn't important, I am only clarifying that if the reasoning is that learning, innovation, and identification of improvements only happen at the Sprint Retrospective, then that is not a good understanding. In fact, nothing stops a Scrum Team from identifying improvements and improving itself or its process from day 1 of the Sprint. The Sprint Retrospective is the formal opportunity for this to happen, in case it hasn't already happened.

Thus, my answer to the question "Which Scrum event is the most important?" is: All Scrum events are important as each of them is an opportunity to inspect and adapt, and increase transparency. This will help enable empirical process control. Period.

This article was originally published at
https://baa.tco.ac/3FD9

About Steve Matthew

Steve has over a decade of professional experience and has worked in various roles, such as Business Analyst, Scrum Master, and Agile Coach.

He has a broad array of real-world experience and has worked in a variety of industries, such as Information Technology, Retail, Banking, and Insurance. He is an Accredited Kanban Trainer (AKT) and SAFe Program Consultant (SPC).

Steve enjoys writing articles and public speaking so that he can help bring awareness within the Agile community on commonly misunderstood or difficult Agile topics.

Ways to Contact Steve

stevematthew911@gmail.com
www.linkedin.com/in/iamsteve

Avoiding The Cobra Effect With Systems Thinking

By Steve Schmitz

Snakes. Why Did It Have To Be Snakes?

"Systems thinking" is a term often used to encourage people to "think of the big picture", but it means so much more than this. Beyond the appeal to be aware of the bigger context, systems thinking is a discipline with a specific set of practices and skills.

Before we get to the snakes, it would be instructive to define "a system". A system is a grouping of certain elements and their interconnections that achieves a purpose.

- *Elements* – The various forces or parts that comprise the system.
- *Interconnections* – The relationships or stories that hold all of the elements together. In systems thinking, the interconnections are more important than the individual elements.
- *Purpose* – The outcomes that the system generates. This is the trickiest to grapple with because the system may produce an overall purpose that is at odds with the purpose of the individual elements.

A marketing department is a system. A river full of flora and fauna is a system. A bowl of Skittles candy is not a system.

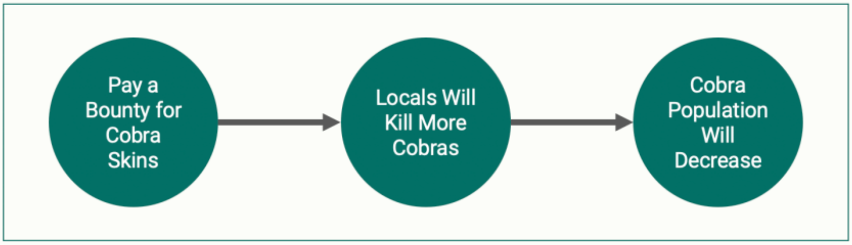
Perhaps the best way to understand what a system is would be to look at one and observe its behavior.

The Cobra Effect

In the late 1800s, the British Empire ruled over colonial India. During this time, the British government leaders became aware of a growing problem in Delhi: Snakes, and not just any snakes, but deadly, poisonous cobra snakes. The population of cobras was growing rapidly and quickly becoming a health hazard for the inhabitants of Delhi.

The solution to the problem seemed clear and obvious: Eradicate the snakes.

The British-led local government set to accomplish this by declaring a bounty on every cobra skin that anyone brought to the government officials. They set up "cobra collection centers" with officials to disburse money in exchange for the cobra skins.

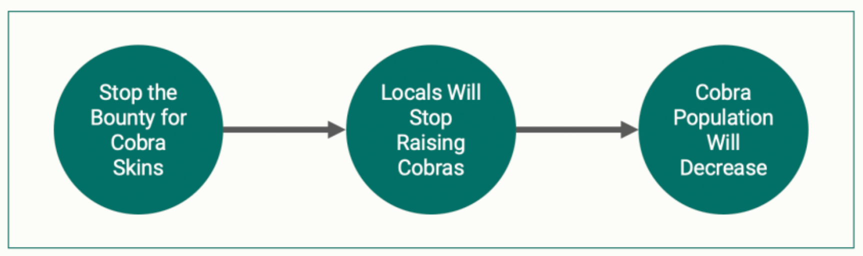


Within a short period of time, the "Cash for Cobra Skins" program appeared to be working. There were fewer reports of deaths due to cobra bites, and there were fewer sightings of cobras in the city.

However, after a while, the situation began to change. The cobra problem resurfaced and began to gradually increase to the point where it was worse than before – much worse. No matter how many dead cobras were collected, both cobra sightings and deaths from cobra bites steadily increased throughout the city. The British were baffled by this turn of events.

Eventually, the British government realized what was happening. They had underestimated the entrepreneurial spirit and creativity of the Indian population. It turned out that shortly after the cobra bounty was announced, the Indians began raising cobras in cobra farms. They had discovered that the amount of money they could make from turning in cobra skins was higher than the cost of breeding and raising cobras to adulthood. Inadvertently, the government had created a brand new cash cow market for cobra skins, and the local Indians who were raising cobras and selling the skins to the British officials were becoming rich.

When the British officials realized what was happening, they immediately cancelled their cobra bounty program to discourage the farming of cobras. Logically, they figured that removing the incentive for raising cobras would reduce the cobra population.



What happened, in fact, was the opposite.

The myriad of cobra entrepreneurs realized that all of their prized cobras in their cobra farms were now worthless. So to minimize their losses, they released all of their cobras into the wild, which caused a new explosion in the cobra population in the city, orders of magnitude worse than the original problem.

Despite the best intentions and efforts of the British government to solve the problem in the city, the outcome was the exact opposite of their intention. Their shortcoming was that they didn't realize they were dealing with a system, so they used a non-systemic strategy to deal with a systemic issue.

The 3 Characteristics of a System

There are generally three high-level characteristics of a system:

1. Systems are *non-linear*. They don't have a starting point and an end point like in a process. Causes and effects don't necessarily flow in a one-directional linear fashion.
2. Systems also generate *feedback loops*. Various elements in the system communicate in multiple ways, meaning that one change in one part of the system can create

ripple effects in multiple parts of the system.

3. Systems also have *emergence*. Even minor changes affecting the system can create increasingly complex behaviors that are much more sophisticated than any one element alone could create.

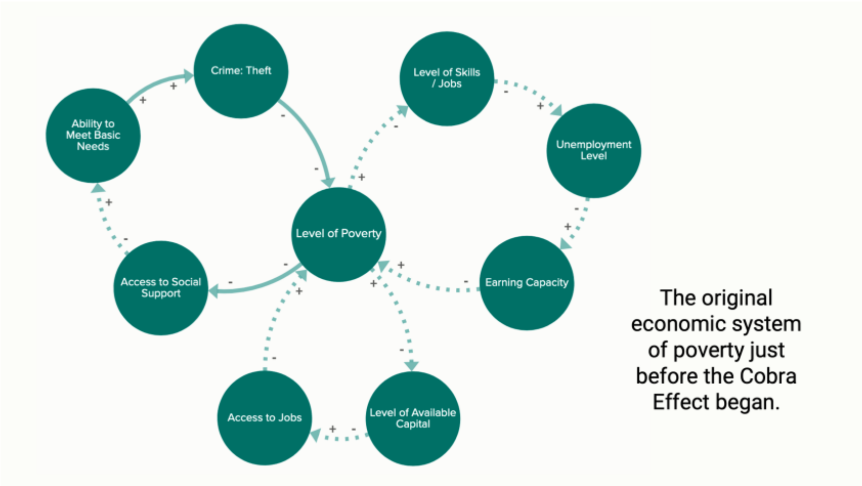
You Know You May Be Dealing with a System If…

There are a few clues you can look for that indicate you may be dealing with a system.

- If you have trouble defining the exact problem, or if the problem seems to shift from day to day or week to week, you might be dealing with a system.
- If your solution fixes your problem one time, but it doesn't solve what appears to be the same problem the next time, you might be dealing with a system.
- If you make a hypothesis that A will lead to B, and instead, A leads to Cleveland, you might be dealing with a system.
- If you're trying to map out the situation in a diagram and you seem to have loops instead of a start and end, you might be dealing with a system.
- If you keep hearing people say, "If 'they' would just do this… it would solve everything" but the problem never gets solved, you might be dealing with a system.

Looking at the Cobra Effect as a System

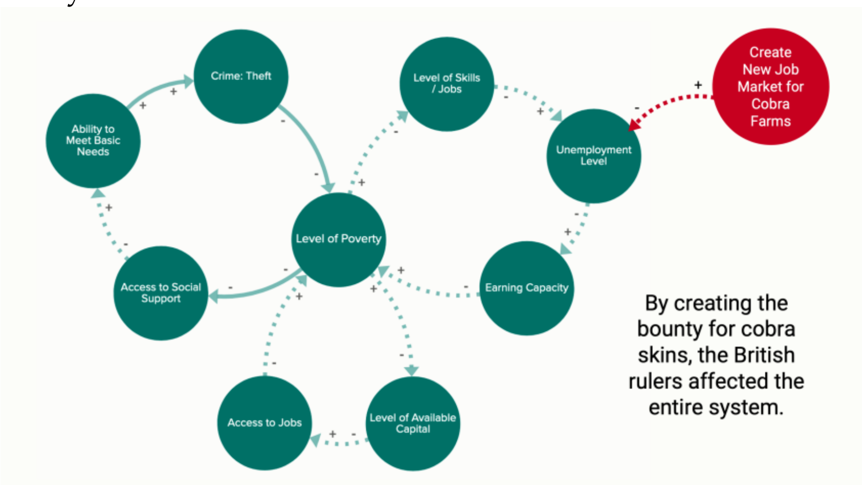
Caveat: Even though I am not an economist, here is a systems map that could be a very simple representation of the economic system that the British officials were unaware of.



The original economic system of poverty just before the Cobra Effect began.

In this system, there is no simple cause-and-effect. There are multiple loops. A change in one element in one loop can reverberate throughout the system and cause unintended consequences in other parts of the system.

The diagram below is the same systems map with the addition of the change that the British officials implemented into the system.



By creating the bounty for cobra skins, the British rulers affected the entire system.

By introducing the bounty for cobra skins, the British officials actually increased the employment level of the economy. This in turn increased the earning capacity of the Indians, which reduced the level of poverty. In turn, this reduction in poverty created ripple effects through the system that were entirely unexpected. One of these unintended ripple effects was a growth in the cobra population, and consequently, an increase of deaths due to cobra bites.

Building Blocks of System

To get a grasp of what is really going on in a complex system, it may help to begin by considering the 5 bulding blocks of any system.

1. What are all of the *elements* within the system? These are the tangible parts of the system that can be physically seen. The elements could be people, machines, offices, materials, manuals, computers, tools, etc.
2. What are the **beliefs** held by the people in the system (assuming that people are a part of the system in question)? Beliefs are more abstract so they might be hidden beneath the surface. Beliefs could be related to values, attitudes toward people or policies, loyalty, resistance, tension, cooperation, etc.
3. What are the *relationships* between the various elements of the system? Metcalfe's Law tells us that the relationships between the elements of a system increase far more than the number of elements in the system. Consider how each element relates to all of the other elements in the system.
4. What are the **loops** in the system? Elements in the system create a life of their own as they interact, forming continuous loops. These loops can be vicious, virtuous, stabilizing, or stagnating. How do elements in one loop feed into another loop? How does one loop as a whole relate to the other loops?
5. What is the *purpose* of the system? It's important to remember that systems are never broken. Systems behave exactly in the way that they were designed to operate. The tricky part is that sometimes, their

behavior is very different than the expectations we have for the system. By understanding how the vicious, virtuous, stabilizing, or stagnating loops in the system interact, one can begin to decipher the true purpose of the system and why it behaves as it does.

Wrapping It Up

Dealing with systems is challenging. Influencing systems in such a way as to achieve desired outcomes in the system is even more challenging. By recognizing a system when we encounter one, understanding the 3 characteristics of a system, and deciphering the 5 building blocks of a system, we can begin to understand the complexity of the system. Then, and only then, can we begin to discover clues on how to apply leverage to help the system change itself to produce more beneficial outcomes.

About Steve Schmitz

Steve is a consultant, trainer, writer, and coach. He has broad experience from garage startups to Fortune 500 companies, private business to government agencies, and agriculture to Silicon Valley.

Since discovering Scrum in the early 2000s, he has helped many organizations find ways to increase the flow of value to their customers. He is passionate about helping people and organizations find innovative ways to improve their service delivery from the customer's perspective.

He holds the Certified Scrum Professional (CSP-SM) and Kanban Coaching Professional (KCP) credentials.

Ways to Contact Steve

ModernKanban.com

steve.c.schmitz@gmail.com

www.linkedin.com/in/steve-c-schmitz

Opportunity Mapping: An Essential Skill for Driving Product Outcomes

By Teresa Torres

The following is an excerpt from my upcoming book, Continuous Discovery Habits. It's the opening to my chapter on Opportunity Mapping. Read to the end for an exciting new announcement.

"To maintain the state of doubt and to carry on systematic and protracted inquiry—these are the essentials of thinking." John Dewey, How We Think

"Structure is complicated. It gets done, undone, and redone."
– Barbara Tversky, Mind in Motion

As you collect customers' stories, you are going to hear about countless needs, pain points, and desires.

Our customers' stories are rife with gaps between what they expect and how the world works. Each gap represents an opportunity to serve your customer.

However, it's easy to get overwhelmed knowing where to start. Even if you worked tirelessly addressing opportunity after opportunity for the rest of your career, you would never fully satisfy your customers' desires.

This is why digital products are never complete.

How do we decide which opportunities are more important than others? How do we know which should be addressed now and which can be pushed to tomorrow?

It's hard to answer either of these questions if we don't first take an inventory of the opportunity space.

A single customer story might elicit dozens of opportunities. If you *interview continuously*, your opportunity space will always be evolving—expanding as you learn about new needs, contracting as you address known problems, and gaining clarity as you learn more about specific needs.

Opportunity mapping is a critical activity. Finding the best path to your *desired outcome* is an *ill-structured problem* and requires that we first structure or frame the *problem space* before we can dive into solving it.

Mapping the opportunity space is how we give structure to the ill-structured problem of reaching our desired outcome.

Mapping the opportunity space is how we give structure to the ill-structured problem of reaching our desired outcome.

It's easy, however, to bounce from one opportunity to the next—reacting to each and every need or pain point we hear about. Most *product teams* are devoted to serving their customers and when they hear about a need or a pain point, they want to solve it.

But our job is not to address every customer opportunity. Our job is to address customer opportunities that drive our desired outcome. This is how we create value for our business while creating value for our customers. Limiting our work to only the opportunities that might drive our desired outcome is what ensures that our products are viable over the long run and not just desirable in the moment.

Our job is not to address every customer opportunity. Our job is to address customer opportunities that drive our desired outcome.

Our goal should be to address the customer opportunities that will have the biggest impact on our outcome first. To do this, we need to start by taking an inventory of the possibilities.

In the quote that opens this blog post, American educational philosopher John Dewey encourages us to "carry on systematic and protracted inquiry." Rather than jumping to the first need that we might address, Dewey argues, good thinking requires that we explore our options—that we carry out a systematic search for longer than we feel comfortable.

We should *compare and contrast* the impact of addressing one opportunity against the impact of addressing another opportunity. We want to be deliberate and systematic in our search for the highest impact opportunity.

In the second quote that opens this post, cognitive psychologist Barbara Tversky reminds us that structure "gets done, undone, and redone."

As the opportunity space grows and evolves, we'll have to give structure to it again and again. As we continue to learn from our customers, we'll reframe known opportunities to better match what we are hearing.

As the opportunity space grows and evolves, we'll have to give structure to it again and again.

As we better understand how our customers think about their world, we'll move opportunities from one branch of the *opportunity solution tree* to another (more on that later). We'll rephrase opportunities that aren't specific enough. We'll group similar opportunities together. These tasks will require rigorous critical thinking, but the effort will help to ensure that we are always addressing the most impactful opportunity.

In this post, we'll tackle the challenges with managing opportunities in a backlog, why tree structures are better for managing the complexity of the opportunity space, and how you can level up your opportunity mapping skills.

Taming Opportunity Backlogs

Some teams are already capturing opportunities in an opportunity backlog. They prioritize their list of customer needs, pain points, and desires the same way they prioritize their user stories in their development backlog.

This is a great place to start. It's better than only working with one opportunity at a time.

However, it can be hard to *prioritize a flat list of opportunities*, because opportunities come in different shapes and sizes, some are interrelated, others are subsets of others.

It can be hard to prioritize a flat list of opportunities, because opportunities come in different shapes and sizes, some are interrelated, others are subsets of others.

For example, imagine you work at a streaming entertainment company and are working with the following list of opportunities:

- I can't find something to watch.
- I'm out of episodes of my favorite show.
- I can't figure out how to search for a specific show.
- I don't know when a new season is available.
- The show I was watching is no longer available.
- I fell asleep and several episodes kept playing.
- I want to watch my shows on my flight.
- I want to skip the show intro.
- Is this show any good?
- I want to know what my friends are watching.
- Who is that actor?
- I want to watch my shows on my train commute.

I don't know how to compare "I can't find something to watch" with "I'm out of episodes of my favorite show." These opportunities are not distinct. Running out of episodes of your favorite show is a reason why you might not have something to watch. But it's not the only reason, so these aren't exactly the same either.

"I want to watch my shows on my flight" and "I want to watch on my train commute" sound similar. Are these really the same opportunity? Maybe they can be combined into, "I want to watch on the go." That might be right.

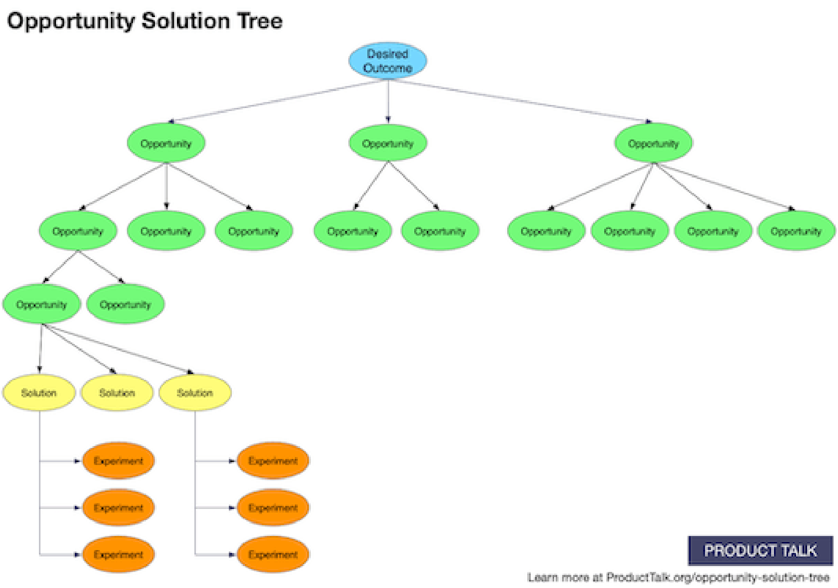
Unless planes and trains introduce different constraints. I may need to be completely offline on a plane, whereas on a train I may still have cell data. I might have access to a power outlet on a plane, but not on a train. If these context differences are important to the experience, these opportunities are similar, but not the same. But how do I prioritize them against each other? Can I address them both at the same time?

"Is this show any good?" feels like a big, hard problem. How do we evaluate "good" for each individual viewer?

"Who is that actor?" feels much easier. Should we always prioritize easy over hard? If so, when do we ever get to the hard problems that have the potential to differentiate us from our competitors and really drive our outcomes?

It's hard to answer these questions when prioritizing opportunities of different shapes and sizes against each other. The opportunity space is too complex to manage as a flat list. Let's turn to a better alternative.

The Power of Trees



Opportunity solution trees help us map out the complexity of the opportunity space.

Instead of managing an opportunity backlog, we'll use an *opportunity solution tree* to help us map out and understand the opportunity space. The tree structure will help us visualize and understand the complexity of the opportunity space.

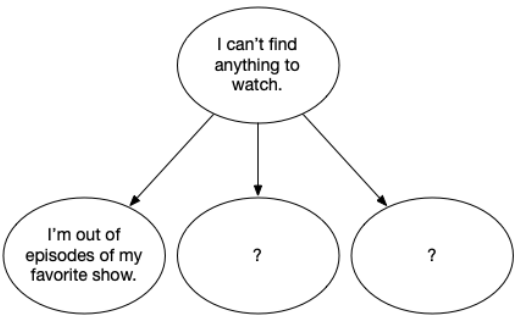
Trees depict two key relationships—parent-child relationships and sibling relationships. Both will help us make sense of the messy opportunity space.

Trees depict two key relationships—parent-child relationships and sibling relationships. Both will help us make sense of the messy opportunity space.

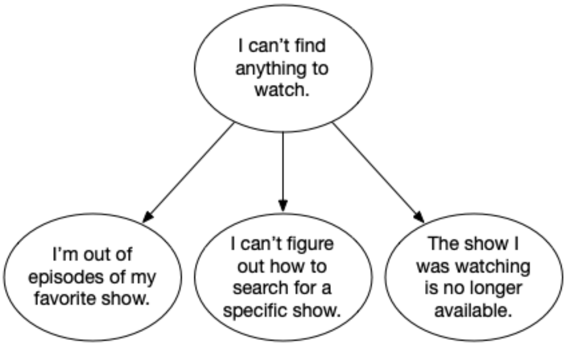
The parent-child relationship will be used to represent subsets—a child opportunity (or sub-opportunity) is a subset of a parent opportunity. For example, in the previous section, we saw that "I'm out of episodes of my favorite show" was one reason, but not the only reason for "I can't find anything to watch."

Referring to the tree relationships, we would say that "I can't find anything to watch" is the parent of the child "I'm out of episodes of my favorite show."

We might then ask, "What are other reasons why I can't find anything to watch?"



We might add, "I can't figure out how to search for a specific show" and "The show I was watching is no longer available" as siblings to "I'm out of episodes of my favorite show."

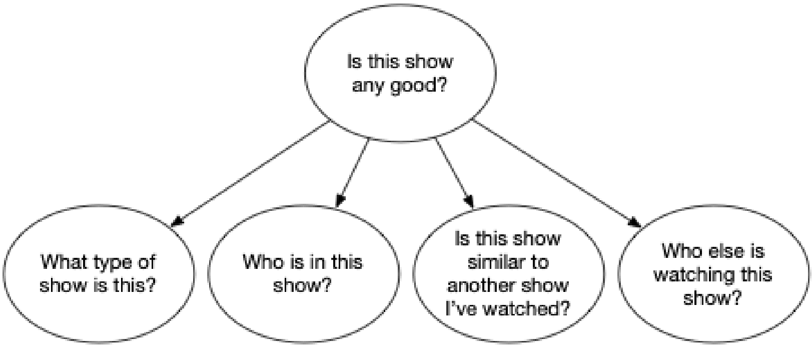


Siblings should be similar to each other—they are each a subset of the same parent, but distinct—you can address one without addressing another. For example, we can address "I can't figure out how to search for a specific show" without addressing "I'm out of episodes of my favorite show." But by addressing "I can't figure out how to search for a specific show," we partially address "I can't find anything to watch."

Sibling relationships help us make sense of similar opportunities like "I want to watch my shows on my flight" and "I want to watch my shows on my train commute."

We can easily depict both on our tree under the parent opportunity "I want to watch my shows on the go." This allows us to treat each context (e.g. plane, train) as a specific need to address, while also visualizing the similarities. They are both sub-opportunities of the same parent.

When we learn to think in the structure of trees, it helps us decompose large, intractable problems into a series of smaller, more solvable problems.



For example, "Is this show any good?" might on the surface look like a challenging problem to solve. But as we dig in and learn more, we realize that different people solve this problem in different ways.

Some people choose what to watch based on the type of show (i.e. they like dramas or crime shows). Others choose shows based on who is in it—they have favorite actors—and they use the cast list as their primary selection factor.

The more we learn about how people evaluate shows today, the more likely we can turn a big, intractable problem like "Is this show any good?" into a series of more solvable problems: "What type of show is this?", "Who is in this show?", "Is this show similar to another show I've watched?", "Who else is watching this show?", and so on.

The big, intractable problem of "Is this show any good?" is a parent opportunity while the rest are its sub-opportunities (or children).

The value of breaking big opportunities into a series of smaller opportunities is twofold. First, it allows us to tackle problems that otherwise might seem insolvable. And second, it allows us to deliver value over time.

That second benefit is at the heart of the *Agile manifesto* and is a key tenet of continuous improvement.

Rather than waiting until we can solve the bigger problem—"Is this show any good?"—we can deliver value iteratively over time.

We might start by solving the smaller problem of "Who is in this show?" because it's fairly easy to solve and because we know a large percentage of our audience chooses shows according to this criteria.

This allows us to ship value quickly.

Now it might not solve the bigger opportunity completely, but it does solve a smaller need completely.

Once we have accomplished that, we can move on to the next small opportunity. Over time, as we continuously ship value, we'll chip away at the larger opportunity. Eventually, we'll have solved enough of the smaller opportunities that we will in turn have solved the larger opportunity.

Additionally, the tree structure is going to be invaluable when it comes time to assessing and *prioritizing opportunities*.

Our goal is to work on the most impactful opportunity, but we can't assess every opportunity we come across. We'd spend weeks assessing the opportunity space instead of shipping value to our customers.

Our goal is to work on the most impactful opportunity, but we can't assess every opportunity we come across. We'd spend weeks assessing the opportunity space instead of shipping value to our customers.

Instead, we can use the tree structure to help us make fast decisions.

While structuring the opportunity space is hard work, the effort is paid back with hefty rewards.

About Teresa Torres

Teresa Torres teaches a structured and sustainable approach to continuous discovery that helps product teams infuse their daily product decisions with customer input.

She's coached hundreds of teams at companies of all sizes, from early stage start-ups to global enterprises, and has taught over 6500 product people core discovery skills through the Product Talk Academy.

Teresa is the author of the upcoming book, Continuous Discovery Habits and blogs at ProductTalk.org.

Ways to Contact Teresa

ProductTalk.org
teresa@producttalk.org
linkedin.com/in/teresatorres/
twitter.com/ttorres

"Product Owner" Is the Most Misunderstood Role in Scrum

By Willem-Jan Ageling

Most adaptations of the Product Owner function invalidate the Product Owner role

Image by Igor Link from Pixabay

Scrum is all about creating products of the highest value in complex product environments. Here's why:

"In today's fast-paced, fiercely competitive world of commercial new product development, speed and flexibility are essential. Companies are increasingly realizing that the old, sequential approach to developing new products simply won't get the job done." — The New New Product Development Game by Hirotaka Takeuchi and Ikujiro Nonaka

If you are unfamiliar with this quote, then it may surprise you that it is from 1986. *It is 34 years old.* It comes from the paper that introduced the first concepts of Scrum: *The New New Product Development Game.*

Since then the world only became vastly more fast-paced, more unpredictable, more complex. Products that used to be popular last year have become obsolete by now. What will be successful next year, nobody knows. With that, long term detailed planning is ineffective. The sequential approach to developing new products has become even more nonsensical than in 1986.

Still, many organisations have not come to realise this. They have tailored Scrum to fit into their traditional processes. The role that suffers the most from this is the Product Owner. Development Teams can have some sort of autonomy in their Sprint bubbles. But Product Owners often are a mere mixture between a Project Manager and a Business Analyst. They ensure that the Development Team knows what to do and aim to follow the Product Roadmap. Rather than actively working to create a product that fulfils the needs of the stakeholders, the Product Owner follows a roadmap defined by others.

By performing these activities, the Product Owner role becomes diluted. As a result, it is a shadow of what it should be. The Product Owner role entails far more responsibilities than many companies believe.

Scrum Guide ambiguity

It doesn't help that the Scrum Guide is ambiguous about the Product Owner role. It starts well:

"The Product Owner is responsible for maximizing the value of the product …" — Scrum Guide 2020

But then it adds:

"… resulting from work of the Development Team." — Scrum Guide 2020

This is where the confusion starts. What does this phrase even mean? It is easy to read this as *'making sure that the Development Team delivers according to a roadmap defined by others, translating the roadmap into the Product Backlog.'* Many organisations work this way. A Product Manager determines the product roadmap. Product Owners update their Product Backlogs to feed their teams with items for their *component*. They don't own a product, they manage a component.

This is also how it often works with SAFe and other similar scaling frameworks.

Companies that work like this ignore the very reason to use Scrum.

The core of Scrum

"The shift from a linear to an integrated approach encourages trial and error and challenges the status quo. It stimulates new kinds of learning and thinking within the organization at different levels and functions." — The New New Product Development Game by Hirotaka Takeuchi and Ikujiro Nonaka

This is another 34-year-old quote that captures the core of Scrum. Empiricism is the foundation of Scrum. This is directly related to 'trial and error'. In the world of empiricism, there is no place for detailed and static long term roadmaps. Product Owners need to find other ways to feed the Product Backlog. They need to maximise the value of the product by making the right calls on what to work on next. Scrum's empiricism is crucial here. Especially the feedback from stakeholders at the Sprint Review.

The importance of the Product Owner

In a complex environment, it is the Product Owner who processes all feedback on the product. She/he does so by translating feedback into a Product Backlog ordered by possible value. The Product Owner needs to ensure alignment with many different stakeholders. They can be customers, users, sales, operations, infrastructure, architecture. She/he also needs to review the timeline, budget, potential capabilities and marketplace of the product. The Sprint Review is the logical place to do all this, but the work of a Product Owner exceeds the world of Scrum.

The Product Owner should drive the direction and growth of the product, with the help of the rest of the Scrum Team and the stakeholders.

Component Owners vs Product Owners

Earlier in the article, I mentioned how Product Managers create a product roadmap. The product roadmap is input for multiple Product Owners and their Scrum Teams. Each Product Owner works on a component of the product. This means that she or he is a *Component Owner,* not a Product Owner. This is a Scrum anti pattern.

I already mentioned that this setup is ineffective in a complex product environment. You can't make detailed long-term roadmaps. But there's more to it. The *Component Owners* may have a good view of their component. But they don't see the total picture of the product, the sum of the components. With that, they are not in a position to decide what item from the component is the most important compared to the other components, or for the product as a whole. Only the person that has the total oversight can do this. In this example, no-one has this oversight. The Product Owner responsibilities are split between the Product Manager and the Product Owners.

Product Owner role vs Product Owner function

Scrum introduced a couple of roles, the Product Owner and the Scrum Master. Quickly organisations started to create job functions called Product Owner and Scrum Master. On a small scale, role and function are often the same. But things often go wrong on a larger scale when more than one team works for the same product. There are two ways to wrongfully creating a Product Owner function. Often you see a combination of the two:

- Some organisations believe that every Scrum Team should have a Product Owner. They have multiple Product Owners per product. Together, they own the product.
- Other organisations have a Product Manager for the end responsibility and Product Owners on 'Scrum level'. Here Product Owners don't even have the final say over their Product Backlogs. This is the Product Manager's call.

Both adaptions of the Product Owner *function* invalidate the Product Owner *role*. In the first example, there should be only one person responsible for the product:

"The Product Owner is one person, not a committee. The Product Owner may represent the desires of a committee in the Product Backlog, but those wanting to change a Product Backlog item's priority must address the Product Owner." — Scrum Guide 2020

Note that there is only one Product Backlog for a product and one person should have end-to-end accountability. One person should have the role of Product Owner.

In the second example, there's one person in a position to have the end responsibility. The Product Manager should understand that she/he should embrace the Product Owner *role*. The people with the Product Owner *function* are important members of the Development Team. They translate the needs and expectations of product into a language understandable for the Development Team. But they don't have a Product Owner *role*.

The power of the combined Sprint Review

There's another related and important topic, the Sprint Review. This is the event where Scrum Teams and stakeholders discuss the current state of the product. This results in clear directions on what to do next. When 10 teams all help to create a product, how are you going to reach all your stakeholders with 10 Sprint Review events? The simple answer is: you won't.

Instead, the 10 teams should have a combined Sprint Review, guided by the person with the true Product Owner role. In my example, this is the person with the Product Manager function. At such a Sprint Review, all teams together discuss with stakeholders.

Only a complete Increment — the latest state of the Product — is valuable to inspect. Also, findings from stakeholders are easily shared and discussed with all involved development teams. 10 separate Sprint Reviews that all tackle specific components can't be this powerful. Also, having one event discussing the product is bound to draw key stakeholders more than the ones touching only aspects of it.

Value the Product Owner!

In complex product environments, a Product Owner and the Development Team(s) should be working together to maximise the value of the product by creating an Increment and inspecting it, learning from it. This does not come across well enough in the Product Owner section of the Scrum Guide. If you isolate this section, you miss out on empiricism completely. What's more, if you only take into account the Product Owner passages of the Scrum Guide, you won't find entries discussing empiricism, the heart of Scrum.

The Product Owner role is open for multiple explanations. As a result, some Product Owners only manage the Product Backlog while other Product Owners are responsible for the product vision and strategy.

A major issue arises when bringing traditional product delivery methods into the mix. Conventional delivery methods do not cater to complexity. It is dangerous to assume that portions of these methods mix with Scrum. Instead, they can lead to disastrous side-effects.

For Scrum to work, realise why it exists. It is there to create products of the highest value in complex environments. To succeed, a Product Owner should be the one with the final voice on what should happen next to the product utilizing a Product Backlog. The Product Owner should own the complete product, not components of the product.

The Product Owner will get the best results in a Sprint Review with all the Scrum Teams working on the product. Such an event will inspect the complete and integrated product Increment. This allows for high-value discussions and decisions.

The Product Owner role is very much misunderstood and misapplied. It is the most underrated role within Scrum. This is a role that in theory entails far more responsibilities than what you see happening in practice.

This article was originally published at
https://baa.tco.ac/3FCU

Don’t Waste Your Sprint Review. Make It Bigger Than Just Your Sprint

By Willem-Jan Ageling

From https://www.pexels.com/@skitterphoto

I love Sprint Reviews. At this event, empiricism works full throttle. The Scrum Team has added new potential value to the product Increment. Now the team and its stakeholders inspect the Increment and discuss what to do next.

The Sprint Review can be a powerful event to influence the direction of the product. But often Product Owners and their Scrum Teams waste excellent opportunities. They are in a vicious circle and only discuss the latest additions to the products. The stakeholders appear out of courtesy, not to help improve the product. In this unfortunate scenario, the Sprint Review is relegated to be just a demo or a status report.

There is more to a Sprint Review than that. The Sprint Review is the place to provide feedback on insights from what the team built. Also, it’s the place for stakeholder feedback on trying the released version of the product.

This is important. You want your Scrum Team to work on the most valuable stuff in the next Sprint. Help them do so with a high-quality Sprint Review.

This is what I am going to address here and now.

Uncovering the potential of the Sprint Review

To uncover the true potential of the Sprint Review, let's start with assessing why it exists:

"A Sprint Review is held at the end of the Sprint to inspect the Increment and adapt the Product Backlog if needed." — Scrum Guide 2017

Many read this quote as:

"We show what we have created during the Sprint and gather feedback from our stakeholders about it".

But this is not what the quote states. Let's unpack this a bit more and see what an Increment really is:

"The Increment is the sum of all the Product Backlog items completed during a Sprint and the value of the increments of all previous Sprints." — Scrum Guide 2017

Many believe the Increment is only the sum of all the Product Backlog Items completed in the Sprint. But that is not the case. The Increment is all existing features plus the new functionality you added during the Sprint. It is the whole product, including what is new. It is the new product update to be released whenever the Product Owner wishes to do so.

When you realize this, you will not limit the discussions to what the team did in the last Sprint. With that, it makes the Sprint Review interesting for a larger audience.

New insights to the product can come from all kinds of places, not only from a demo during the Sprint Review.

Here are some examples of what I mean:

- The team repaired errors in the displayed calculations. Only a small part of your target audience will be interested in seeing this. If your Sprint Review focuses on this only, it falls short on what it could establish.
- The work from the Sprint can have an impact on other — already released — functionality. This without you realising. By demoing and discussing the complete product Increment you can uncover this. Stakeholders may have insights into the impact you never realised before.
- A stakeholder wants to discuss functionality created a few Sprints ago. They recently received positive customer feedback about it. This may be important to further uncover and exploit.
- A stakeholder wishes to discuss functionality created a few Sprint ago. Only now it turns out to be faulty.
- There are new market developments and the product needs to be able to cope with them. A new payment method is becoming very popular and your product needs to be able to use this method to stay relevant.

At the Sprint Review, discuss everything new about the complete product Increment.

In a complex environment, it is important to factor in all new information. It can all impact the decisions on what to do next. Scrum Teams work in complex environments. When you get a good sense of how the increment is working and respond to new information in a Sprint Review, you are adopting the best available tactic in this kind of environment.

"What has happened should be used for forward-looking decision-making." — me, slightly adjusting a quote from the Scrum Guide 2017

Lessons for an effective Sprint Review

When you ensure the discussions tackle the complete product Increment you allow for strategic product decisions. As a result, the Sprint Review becomes important for senior stakeholders too. The Sprint Review turns into the Steering Committee for the product.

I worked as a Scrum Master for many teams over the years. I have identified two lessons with true stakeholder involvement at the Sprint Review:

Lesson 1: Discuss the complete product Increment. This increases the chances of getting important stakeholders to join the event as they can bring meaningful insights to the table.

Lesson 2: Ensure your essential stakeholders attend the Sprint Review. Allow them to discuss the complete Increment. It will turn the Sprint Review into the event where major product decisions happen. These product decisions impact the Product Backlog.

Both lessons strengthen each other!

Below are three real-life cases to clarify the lessons.

Example 1 — Discuss the complete increment

We made sure to discuss the complete product Increment all the time. At first, only the Scrum Teams and a portion of the stakeholders participated. But soon word spread how impactful the discussions about the product were. Then more stakeholders joined. This included people that were several layers higher in the organisation and people that used the product daily. The Sprint Review made a separate product roadmap meeting redundant. All potential participants for such a meeting were in the Sprint Review already. Decisions didn’t need another discussion.

Example 2 — Essential stakeholders attend the Sprint Review

Another team built a fraud detection product from the ground up. Right from Sprint 1 all essential stakeholders — including the users and directors — were involved in the Sprint Review. We looked at what the team built during the Sprint, new ideas and the actual usage of the product until now. We did not limit the discussion to only include work completed during the most recent Sprint. The Sprint Review was where we had everyone in the discussion. Based on the Sprint Review results, we always could work on what mattered most. The financial regulators of the Netherlands considered our fraud detection product to be best in class.

Example 3 — Failing to keep the momentum

A third team started their journey well. They had a daunting objective. They needed to create a new product using promising but unproven technology. The first Sprint was all about showing they could create such a product. So they built a rudimentary MVP. They succeeded to do so and the Sprint Review was a major success. But after this, the team went on to only demo the new functionality. More and more stakeholders skipped the event. They weren't interested in these updates and missed the big picture. After a few Sprints, the team hardly got something useful from the Sprint Review.

Scrum Patterns and feedback beyond the Sprint

I love the Scrum Patterns. They are a set of best practices that help explain the rationale behind Scrum. Whenever I stumble upon something new for me as a Scrum Master, I tend to inspect the Scrum Patterns for more insights. I failed to find this for my experiences about the Sprint Review. But, I found a snippet that appears to contradict what I experienced:

"Many Scrum adherents view the Sprint Review as the main

mechanism of agile feedback in Scrum, bringing to mind the usual forces of emergent requirements and market changes, change in business conditions, and so forth. There may be a bit of that. But a Product Owner and other stakeholders are unlikely to always be able to assess, in a three-hour meeting (isn't it 4? — WJA), whether a complex product increment really does meet the intended need or not. More realistic and honest insight comes from trying to use the product in a more realistic application setting, over a more protracted period of time." — A Scrum Book, Sprint Review

I agree that you can't assess everything about the product in the timespan of a Sprint Review. But the most time-consuming parts (building the product Increment, gathering feedback from product usage) happen before the event. The Sprint Review is the place to share this feedback and then discuss insights based upon the observations. I argue that there's plenty of time to do this at this event.

The Sprint Review is the place to provide feedback on insights from also *trying the product over a protracted period of time*. If the Sprint Review isn't, there needs to be another meeting. I believe this would be unwise:

"Prescribed events are used in Scrum to create regularity and to minimize the need for meetings not defined in Scrum." — Scrum Guide 2017

Maximize the value of the Sprint Review

In a complex product environment, you base decisions on what you know. Even though you only have fragmented knowledge. Sometimes information about a certain change only arrives 3 months later. Don't let it stop you from bringing it up at the Sprint Review.

There shouldn't be any 'we already passed this stage' mentality. That *is traditional waterfall thinking.*

Scrum Teams work empirically. This means we learn from new information, and use it to influence ideas about next steps. Regardless where it comes from.

A Sprint Review can last 4 hours. Often though, the event is way shorter as it is a demo only. Perhaps, the team gathers some feedback about this demo. Teams that do it like this waste an opportunity to gain the latest insights on the product. They fail to have the best possible view on what to do next.

Shift the focus from output to outcomes. The Sprint produces an output, the outcome may come later. By shifting focus you can discuss the actual impact of your product.

- Discuss the complete product Increment. Include the new learnings from the functionality you built and shipped earlier. It will allow stakeholders to bring information about the product to the table. Topics like actual usage or user experience.
- Have essential stakeholders at the Sprint Review. Allow them to discuss new learnings about the complete product Increment. The Sprint Review could then make other strategic product meetings obsolete.

These two practices strengthen each other.

These lessons may impact the way you facilitate the event. You may divide the event into slots to allow people to attend the parts they wish to be at. Who says everyone needs to be there for the whole four hours? Having said this, the best decisions come with all the stakeholders together. But, there are options. There's no one good way.

This article was originally published at
https://baa.tco.ac/3Eln

SAFe's Scrum vs Scrum According To the Scrum Guide — They Are Not the Same

By Willem-Jan Ageling

A comparison of two different beasts

The Scaled Agile Framework (SAFe) exists to *"to enable the business agility that is required for enterprises to compete and thrive in the digital age."— scaledagileframework.com/about*.

Organisations adopt SAFe to align multiple teams working on the same solution or program. SAFe is one of *many ways to scale*.

The *Agile teams* in a SAFe environment can make their own choices from the many Agile practices to deliver valuable software:

"SAFe teams use Agile practices of choice based primarily on Scrum, Kanban*, and Extreme Programming (XP) to improve their performance." — SAFe 5.0/Agile Teams*

Many Agile teams choose to work with Scrum. However, the above link doesn't point to the *Scrum Guide version of Scrum*. It points towards a SAFe page discussing "ScrumXP". This is a hint that ScrumXP is the preferred way of working within SAFe.

ScrumXP *is the standard, adapted version of Scrum in a SAFe environment.*

ScrumXP differs from Scrum according to the Scrum Guide. Often this is due to elements of XP, sometimes not. It is true that Scrum is an add-on framework. Introducing XP practices in Scrum is totally in line with Scrum and often happens. However, SAFe takes it further than adding XP to Scrum.

SAFe combines Scrum and XP frameworks to create its own ScrumXP process as part of its framework. SAFe gives Scrum and XP terminology new meanings and structure. This reduces the transparency over what Scrum (and XP) is. This article aims to clarify to practitioners the differences between Scrum as defined in the official Scrum Guide and ScrumXP as defined in SAFe.

Image by Arek Socha from Pixabay

Definition and usage

"ScrumXP is a lightweight process to deliver value for cross-functional, self-organized teams within SAFe." *— SAFe/ScrumXP*

SAFe uses Scrum to guide team agility and XP for technical guidance.

Scrum has a different purpose than ScrumXP:

"A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value." — *Scrum Guide 2017*

I find it interesting to note that the Scrum Guide says this:

"Scrum is not a process, technique, or definitive method. Rather, it is a framework within which you can employ various processes and techniques." — Scrum Guide 2017

There's an important difference between a process and a framework. A process is more directive than a framework. As a result, the different ways to look at ScrumXP and Scrum say something about the use of the approach.

Scrum is a framework to tackle complexity and deliver value while ScrumXP is a process to deliver value with SAFe. This is not the same.

ScrumXP is something you run as a container in SAFe. It's a process within SAFe and not supposed to change how you work.

Composition and characteristics of the team

While Scrum has Scrum Teams, Scrum XP has Agile teams. An Agile team using ScrumXP self-organises, self-manages and is cross-functional. Scrum doesn't mention self-management. Both approaches stress the importance of teams finding their own best way to deliver value, based on the intent or the goal. Both also acknowledge that having this team composition can help raise productivity and enjoyability.

Product Owner and Scrum Master

ScrumXP has two special roles: Scrum Master and Product Owner. The Scrum Master focuses on the effective use of ScrumXP and on removing impediments. The Product Owner focuses on what the Agile team will build.

Both roles in ScrumXP deviate a lot from the Scrum role. You can read more on this in the detailed articles discussing the *Scrum Master* and the *Product Owner*.

The Sprint/Iteration

SAFe uses a different word for the timebox in which the team builds an Increment: Iteration. This is just like *XP*. Apart from that, the objectives of an Iteration are the same as the objectives of a Sprint in Scrum: producing an Increment that delivers value.

Sprint/Iteration Planning

The length of a ScrumXP Iteration planning is four hours or less. The Scrum Sprint Planning is eight hours for a one month Sprint and less for shorter Sprints. Both with ScrumXP and with Scrum, the Product Owner brings forward what should be achieved and the team decides how much they can do and how they will build it.

Both planning events revolve around defining a goal for the Iteration/Sprint and selecting Stories/Product Backlog Items. ScrumXP starts with creating an Iteration Backlog and then defines an Iteration Goal. *Teams commit to meeting the goal and finishing the Stories on the Iteration Backlog.*

With Scrum, however, teams first discuss the Sprint Goal and then creates a Sprint Backlog, which has the selected Product Backlog Items to meet the Sprint Goal and a plan for delivering the Increment and meeting the Sprint Goal. *Teams commit to finishing the Sprint Goal and the Sprint Backlog is a forecast only* as it can change to allow inspection and adaptation to optimise the chances to meet the Sprint Goal.

As you can see, there are pivotal differences between ScrumXP's Iteration Backlog and Scrum's Sprint Backlog.

But ScrumXP teams commit to more:

"Some teams break each story into tasks. […] As team members commit to tasks, they reduce their individual iteration capacity until

it reaches zero." — SAFE5.0/Iteration Planning

Here ScrumXP even discusses commitment to individual tasks. Scrum also mentions breaking up work in units of one day or less. However, teams or individuals do not commit to them. This would violate the rules of Scrum where the Sprint Goal is the true north of a Sprint, not the Sprint Backlog.

ScrumXP has detailed guidelines on how to facilitate an Iteration Planning, like:

- An Iteration Planning agenda;
- How to estimate (using story points);
- How to establish capacity.

Scrum does not have these details. It merely discusses that Sprint Plannings have two parts:

- What can be done?
- How will it be done?

It is up to the Scrum Team to determine how to do it.

Daily Stand-Up/Daily Scrum

SAFe again uses the XP term for the daily alignment event. However, this doesn't change the purpose of the event, to understand how the team is progressing towards the Iteration/Sprint Goal and adapt if this increases the chances of meeting the goal. SAFe also adds some XP practices to the event. With that, the event is more prescriptive than the Scrum Guide equivalent.

In SAFe, the Scrum Master ensures that the event doesn't take more than 15 minutes. The Scrum Guide only brings forward that the Scrum Master *teaches* the team to do this.

Iteration Review/Sprint Review

With SAFe, the Iteration Review is a demonstration to the stakeholders of the things they completed. It is an opportunity to inspect if the Increment is working according to stakeholders expectations and to adapt if this is needed. The team also discusses progress towards the Program Increment objectives. The timebox is one to two hours.

Scrum's Sprint Review, however, entails much more. This Scrum event with a maximum timebox of 4 hours is an opportunity to inspect if the product is heading in the right direction and discuss what to do next, adapting the Product Backlog if needed:

"The entire group collaborates on what to do next, so that the Sprint Review provides valuable input to subsequent Sprint Planning;

Review of how the marketplace or potential use of the product might have changed what is the most valuable thing to do next; and,

Review of the timeline, budget, potential capabilities, and marketplace for the next anticipated releases of functionality or capability of the product." — Scrum Guide 2017

SAFe pushes this to the *Inspect & Adapt event*, at the end of the Program Increment. With that, an important aspect of Scrum's empiricism falls outside of a Sprint/Iteration and outside of Scrum. It delays feedback, impacting business agility.

Iteration Retrospective/Sprint Retrospective

This event basically has the same objective for both ScrumXP and Scrum. ScrumXP puts forward that the timebox should be one hour or less. Scrum discusses that it should be three hours or less.

ScrumXP is more prescriptive than Scrum, but this should be no surprise anymore. Other than that, there's not much difference.

Not in ScrumXP

The Scrum Guide has items that are not in ScrumXP. Here follows a list.

Scrum Values

Scrum puts the spotlight on the values of commitment, courage, focus, openness and respect. ScrumXP doesn't have these values. SAFe has four values: alignment, built-in quality, transparency, and program execution. These are by no means the same as the Scrum Values.

Empiricism

ScrumXP doesn't mention empiricism. The Scrum Guide considers empiricism to be the foundation of Scrum. It is a vital piece of the puzzle to understand Scrum.

The main article of ScrumXP also doesn't mention transparency, inspection and adaptation, the three pillars of empiricism. Inspect and adapt are terms that you will find other pieces of the SAFe puzzle, but you will hardly find it in the context of ScrumXP.

By removing the emphasis on empiricism, ScrumXP indeed becomes a process to deliver product Increments. Whether they are of value is addressed later, at the end of the Program Increment. ScrumXP is not a framework to address complex products like Scrum is.

Agile Release Train

ScrumXP also has topics involving the Agile Release Train and coordination between teams. I think it is obvious that Scrum doesn't discuss this.

Endnote

ScrumXP, SAFe's delivery process inspired by Scrum and

Extreme Programming, is not the same as Scrum as defined in the Scrum Guide. ScrumXP isn't the typical blend of Scrum and XP.

Most notably, ScrumXP doesn't mention the Scrum Values and Empiricism, both pivotal for Scrum.

An Agile team, the ScrumXP equivalent of Scrum Team, is part of a bigger picture. Their prime objective is to deliver Increment of valuable products in line with the Program Increment Objective. A Scrum Team has a different purpose, addressing complex products while delivering value.

SAFe is the framework and ScrumXP is a process, part of this framework.

As a result, ScrumXP has a different expectation from an Agile team, Product Owner and Scrum Master.

All in all, ScrumXP and Scrum are totally different. This should be no surprise as they serve different purposes.

This article was originally published at
https://baa.tco.ac/3ExQ

About Willem-Jan Ageling

Willem-Jan Ageling is co-founder, editor and writer of Serious Scrum, the fastest growing publication about Scrum and Agile on Medium and independent community for Scrum practitioners.

As a writer for Serious Scrum, holding the PSM III certificate, Willem-Jan helps to raise understanding of Scrum, aiming to help people to use Scrum effectively.

At his daytime job for Worldline, Willem-Jan is an Agile Coach, facilitating the journey to improve value delivery continuously inspired by Spotify, Scrum and DevOps.

Ways to Contact Willem-Jan

medium.com/@WJAgeling

ageling@gmail.com

linkedin.com/in/willemjanageling

twitter.com/WJAgeling

Agile May Be Key for Companies Experiencing VUCA

By Zak Meziane

During this time of uncertainty, businesses are trying to adapt to the new normal. Here's why the Agile philosophy is suitable for unforeseen business obstacles.

Managers and business leaders have been using the acronym VUCA (volatility, uncertainty, complexity and ambiguity) to describe the uncontrollable aspects of the business environment for years. But now more than ever, the entire world is currently experiencing VUCA on a large scale due to the unprecedented impact of the coronavirus pandemic. Because of this, many businesses across industries are struggling to understand the implications of the shift in work and how to adapt.

While there has been a myriad of tips on best practices for a remote structure, business leaders should also consider embracing an Agile mindset to leverage collaboration while we're all working differently. Agility is known to be particularly suitable for complex work with a high level of uncertainty. In fact, the very principles of Agile require adaptation to new variables that can change the course of a project.

Image: rawpixel - stock.adobe.com

Agile was originally created for software development teams in 2001 to help companies innovate faster while keeping up with customer needs. Now, companies beyond IT are embracing the Agile philosophy and finding success in breaking down organizational silos, boosting cross-functional collaboration with significant results. Organizations have seen improved customer satisfaction by delivering more value, faster. After all, "delighting the customer" is the No. 1 principle of the Agile manifesto.

In the high levels of VUCA we're currently facing, there are no standard "best practices." I am currently supporting an Agile transformation for a large financial services client and have found through my coaching that solutions will need to be *discovered* rather than predetermined. In order for companies to find an effective resolution to these complex problems, adopting an Agile method, such as Scrum, can help create an environment where solutions can be discovered rapidly. Here are two ways Agile practices may help your company during our current environment:

1. Agile provides clarity over direction

When coaching organizations through their Agile transformation journey, I spend a great deal of time with senior and middle management. While it's up to the entire company to transform, management has a unique opportunity to shape the future of their organization by role-modeling the behaviors that create the environment in which Agile practices are most likely to flourish. The biggest opportunity for development is in the mindset shift from "command and control" to "servant leadership." In the servant leadership model, the hierarchy of an organization is turned upside down to put employees at the top while leaders are charged with serving those employees above them.

In this time when most are working remotely, the idea of being close to the details is removed -- allowing leaders to slow down and look at the bigger picture. This is instrumental in terms of reflecting on the overall direction of a project, supporting teams from the bottom of the hierarchy and letting them get to the endpoint on their own.

When we apply the three pillars of Scrum -- transparency, inspection and adaptation -- leaders in a Scrum environment have real-time visibility of progress and impediments during each sprint or breakdown of a project. This can help alleviate any anxiety that management may feel as managers move from a control mindset to a service mindset.

2. Agile creates an environment valuable to the team and customers

While in a state of VUCA, enterprises need to be resilient in order to succeed. I have found the best way to be resilient in times like these is to support Agile behavior among the entire company. Agile organizations tend to have environments and cultures that support quick decision-making, creativity in the face of new unprecedented challenges and psychological safety for those who dare to try something new in the hope of solving a customer problem. It also enables teams to change direction quickly to identify new goals, challenges and chances. To achieve this, leaders must understand the importance of experimentation and adaptation. This will further encourage employees to think and behave in innovative ways as opposed to demanding solutions on a timeframe that realistically won't produce end value for customers.

Understanding that these times will bring about many new obstacles, being equipped with an Agile mindset of "failure recovery" instead of "failure avoidance" will create a healthy environment across the organization. For example, when an issue occurs in a highly Agile environment, all parties will work together to solve it. Through this philosophy, organizations can prepare their teams to be open to uncertainty, allowing rapid and effective resolutions to arising problems.

As a Scrum Alliance Certified Agile Coach, I have been helping many organizations transform to Agile while also shifting to a remote workforce. While there is no one-size-fits-all solution for what organizations throughout the world are collectively experiencing, it is worth noting that Agile is most suitable for complex adaptive work where there are relatively high levels of VUCA. When organizations are Agile, they allow for the inevitable testing-and-learning phase that organizations must face. This is because Agile allows leaders to express clarity over direction and work in an environment where the people who are closest to the customer are trusted and empowered to serve them in a way that creates value for all.

About Zak Meziane

Zak is an Agile Transformation & Leadership specialist, with certifications including Certified Enterprise Coach (CEC) and Certified Team Coach (CTC).

Based in London, United Kingdom, he has over 15 years of experience in helping organisations and their leaders thrive in a complex and changing economy by transforming the way they think and work.
Zak has supported organisation-wide agile transformation programs for blue chip organisations such as Bank of Ireland, Direct Line Group and bp, coaching at leadership and team level.

Ways to Contact Julee

www.zakmeziane.com
zak.meziane@katalusis.com
www.linkedin.com/in/zakmeziane

Good Leader to Great Leader

By Zeeshan Amjad

Let me start a discussion with a beautiful sentence written on the preface of the Agile Leader Toolkit book [1]. "A Leader is like a farmer, who doesn't grow crops by pulling them but instead creates the perfect environment for the crops to grow and thrive." Everyone has a different definition of leadership, and that also changes over time, Peter G. Northouse explains a brief history of the evaluation of leadership definition[2]. Here is a simple description of leadership from the same book.

"Leadership is a process whereby an individual influences a group of individuals to achieve a common goal." [2]

Here the critical points are "process," "influences," and "common goal." The process makes leadership an interactive event. There is no leadership without any influence, and without any common goal, there is no direction.

Some so many different models describe leadership and leaders. Paul Newton explains ten most popular leadership theories in his book [3], perhaps Tannenbaum-Schmidt seven steps linear model is one of the most cited models explain leadership style from "Tells, Sell, Suggest, Consults, Joins, Delegates" to all the way "Abdicates." John Maxwell 5 level leadership model is also compelling [4] that includes "Position, Permission, Production, People Development," and "Pinnacle" levels. Bill Joiner and Stephan Josephs define five levels of mastery, "Expert, Achiever, Catalyst, Co-Creator and Synergist" with four competencies. [5] David Rooke and William R. Torbert defined seven transformations of leadership "Opportunities, Diplomat, Expert, Achiever, Individualist, Strategist, and Alchemist" their characteristics, weakness, and steps to move to the next level. [6]

No matter which model you follow or where you are in the leadership journey, as Tony Stoltzfus said in his book [7] "To grow as a leader, a person must gradually increase in ability to take initiative and bear responsibility." We don't want to start a discussion about "Manager vs. Leader," we assumed that readers already know the difference; instead, we are going to discuss the simple "Good to Great" approach, as suggested by Geoff Watts [8][9]. We will look at the four different dimensions of a leader, "Coach, Care, Cultivate, and Change" let's call it the leadership 4C model and 3 factors in each dimension.

Coach

Coaching is unlocking people's potential and what else is a better tool than coaching to grow the people. It is one of the most powerful tool leaders has to not only grow people but also create more leaders who can also develop leaders.

A good leader has great answers. A great leader has great questions.

A good leader grows the people by mentoring; however, a great leader grows the people by coaching. Mentor, with his experience, education, and/or expertise, has excellent answers and solutions for problems; on the other hand, the coach has great questions to help people to grow themselves.

A good leader knows how. A great leader knows why.

Simon Sinek gave a great Ted Talk in 2009 and had a book on the same topic [10]. Good leaders solve how to solve the problem. However, a great leader said why we need to solve this problem.

A good leader solves your problem. A great leader will grow you to solve your problem yourself.

A good leader solves your problem, but instead of solving the problem, a great leader solving your problem, grow you that you will be able to solve your problem. As there is an old saying, "Give a man a fish, and you feed him for a day. Teach a man to fish, and you feed him for a lifetime."

Care

Empathy is one of the main attributes of any leader, and leaders show great care and compassion. Without compassion, it is tough to create influence, and without influence, there is no leadership.

A good leader listens to you. A great leader listens for you.

There is no better way to show empathy than listening, and any leader does precisely that. What makes great leaders different than good leaders is, great leader, not only listen to you but also listen for you. In other words, a great leader keeps an eye and ear open for any opportunity for your growth, be it feedback, challenge, or even another prospect.

A good leader solves problems. A great leader convert problems into opportunities.

We already discuss that good leader solve the problem and great leader grow you to solve your problem yourself. A great leader doesn't stop there but also turn the problems into opportunities. Just imagine how the prominent stockbroker can make the big money if there is no volatility in the stock market?

A good leader stands with you. A great leader will put you in front in case of success and behind in case of failure.

Leaders stand with you to show care, compassion, empathy, and concern. However, one of the primary purposes of the great leader is to grow the persons around him/her, and to do this; all great leaders put you in front of success to give you limelight and behind in case of failure to protect you.

Cultivate

Cultivation means foster for growth, and it is one of the main attributes of a leader. I'd say if a leader has all the characteristics without cultivation, then he can be anything but not a leader.

A good leader creates followers. A great leader creates more leaders.

One of my favorite leadership books, which describes the true story, is Turn the Ship Around by David Marquet [11] explain this concept very well to create more leaders, not followers. Great leaders know that creating more leaders, not followers, will keep this cycle continue, and this is what distinguishes him from good leaders.

A good leader motivates you. A great leaders create the environment which motivates you.

I am yet to meet with any leader who is not a motivator. What makes great leaders, apart from a good leader is that great leaders don't need to motivate you, he/she creates an environment that motivates you. As a result, people always feel motivated even in the absence of a great leader because of the environment he/she created.

A good leader gives feedback. A great leader receives feedback.

Everyone needs the feedback to grow, and good leaders do precisely the same for your growth. In addition to that, great leaders also know the importance of receiving feedback to grow themselves too.

Change

Leaders create change and leave the mark, no matter how big or small that difference and mark is.

A good leader focuses on the outcome. A great leader focuses on the vision.

During one of the speaking, I mentioned, the manager focuses on the output; the leader focuses on the outcome, and the great leader, sometimes also known as visionary, focuses on the vision. Vision gives us direction.

A good leader does things right. A great leader does the right thing.

Any good leader not only does the right things but also mentor you to do the right things. On the other hand, great leaders know what is essential to work on that makes him more impactful.

A good leader transforms persons. A great leader transforms societies.

I keep the best one to the last. No doubt, a good leader transforms and grows the people, but a great leader is not limited to people he/she transforms the whole societies and sometimes even a reason to create new societies. Remember Nelson Mandela and Martin Luther King Jr, not too far from history.

Reference

1. Agile Leadership Toolkit by Peter Koning
2. Leadership: Theory and Practice seventh edition by Peter G. Northhouse
3. Leadership Theories by Paul Newton
4. 5 Levels of Leadership by John Maxwell
5. Leadership Agility by Bill Joiner and Stephen Joseph
6. Leadership Coaching by Tony Stoltzful
7. Seven Transforamtion of Leadership by David Rooke and William R. Torbert
8. Scrum Mastery: From Good to Great Servant Leadership
9. Product Mastery: From Good to Great Product Ownership
10. Start with Why by Simon Sinek
11. Turn the ship around by David L. Marquet

This article was originally published at
https://baa.tco.ac/3ENF

About Zeeshan Amjad

Zeeshan Amjad is a lifelong learner. With over 20 years of experience in Information Technology, he played several roles as Agile Coach, Agile Transformation Leader, Tech Evangelist, Development Manager, Project Manager, Enterprise Architect, Team Lead, Scrum Master, SAFe Agilist, Mentor, Speaker, and Author.

He enjoys Scrum to solve complex problems (not only in Software but also in other domains) and Coaching to grow people by unlocking their own potentials. He loves reading, writing, public speaking, and have a healthy discussion about coaching and Scrum.

Ways to Contact Zeeshan

mzeeshan.amjad@gmail.com

www.linkedin.com/in/thezeeshanamjad

Coaching Human Behaviours

By Zia Malik

You've been there before, you're dreading the meeting, it's 10 a.m. on Monday and you're at your weekly review session with stakeholders. In most cases, it would be fine, in this case, however, it isn't. You can't shake the deep feeling in your stomach, that feeling of innate fear. "I don't want to be at this meeting," you repeat your mantra as you enter the room. You look for one of the stakeholders, the one that completely dominates the room, belittles everyone in your team and rather than celebrate success, victimises and leaves your team broken. It's a repeating pattern, and to be frank, it's bringing you and your team down. Like a virus spreading through your organisation, you notice these behaviours infecting more and more people and you’re not sure what to do about it. You see these extreme behaviours becoming the norm, they've become the personality of your organisation, the silent expectation, the default position.

Sounds familiar? Time and again, as Certified Agile Coaches we find ourselves observing these behaviours. Yet, often we cannot see the best way to navigate through the complex interpersonal relationships to know where they should start, let alone how we might go about dealing with some of these challenges. During organisational change initiatives, we find the emphasis placed on process and structure of change rather than individuals and their place within and the impact on the organisation.

The focus on human behaviours, how people interact and their impact on one another is seldom the intent for an organisational change initiative. Even well-meaning coaches and consultants will often come armed with oven-ready frameworks and models to implement. The impact can be far more damaging to an organisation than one realises.

"If organisations truly want to maintain the effect of change in a continually evolving environment, it has to do something different. And that starts with its people."

Ineffective apathetic teams, despite the effort on team formation and tooling, are often the cause of unengaged people in the workplace, despite leadership's best intentions. If the leaders have placed a great deal of emphasis on building a collaborative and fun organisation, why are they not realising value? Even though they've employed an army of coaches and consultants to help bring their expertise on organisational change to help get resources (sorry, I mean "people") together into a new way of working? Of course, it's all important, but this lack of a holistic focus on organisational change completely misses the point. It creates half-baked change initiatives that are as fragile as the paper they're presented on, taking only the smallest challenge for people to go back to their default positions. Change becomes superficial at best, and its impact is barely felt across the organisation. If organisations truly want to maintain the effect of change in a continually evolving environment, they have to do something different. And that starts with effective coaching of their people.

"As coaches, we are present to encourage a heuristic approach to change. Fundamentally, that starts with human behaviours. If we can't get that right, we are unfortunately building everything else on crumbling foundations."

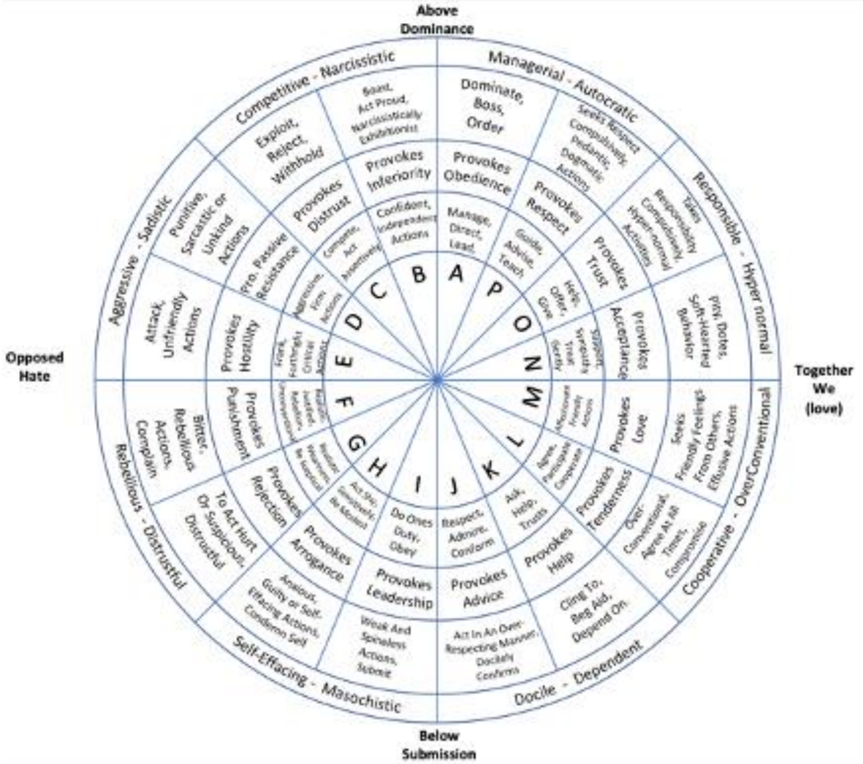
Holistic coaching is close to my heart and something I speak about a lot. All too often, I hear people say, "I'm an executive coach," "I'm coaching the executives," or "coaching the teams," completely ignoring everyone else in the organisation. As a

Certified Agile Coach, I have never decided that I am coaching only a particular group or type of people within the organisation based on structure or hierarchy. Nor do I care for appeasing hierarchical structures by differentiating between a team or executive coach. For me, it's about coaching people full stop. I often call myself an enterprise or organisational coach for this reason; it's about coaching the whole system to change.

Of course, it starts with people, and no matter where we begin, if we don't consider the interpersonal relationships that exist in the organisation, then we will struggle, as coaches, to make any meaningful impact. As coaches, we are present to encourage a heuristic approach to change. Fundamentally, that starts with human behaviours, if we can't get that right, we are unfortunately building everything else on crumbling foundations.

One model that's stuck with me for many years is Timothy Leary's circumplex often referred to as 'Leary's rose.' Timothy Leary was a clinical psychologist and in 1957 wrote a paper in which he explained that interpersonal relationships could be mapped and defined in a circumplex. The circumplex is arranged with extreme behaviours on the outer edge that are divided into segments. The closer you identify to the centre of the circumplex, the more you exhibit well rounded, positive behaviour with others. For me, the circumplex helps coaches recognise negative behaviours and approaches for dealing with them. The top and bottom of the circumplex are considered triggers, so dominant behaviour will encourage submissiveness and vice-versa. On the right, togetherness will promote friendship and support while on the left-hand side, the opposition will encourage conflict.

Leary's Rose



The circumplex is extremely deep, and despite being introduced to me in 2012, there is still much more to learn from and experiment with. The intent of this article is not to describe the detailed patterns associated with Leary's rose; perhaps I will get to that another time. It is to introduce and entice you to learn more about it, given many models come and go and this one has held firm for over 60 years! For organisational coaches, this model can prove to be extremely useful to support deep-rooted organisational change. So many people in organisations suffer from a lack of psychological safety, limiting beliefs, and daily emotional challenges that no amount of process-focused frameworks will help. Coaches need to understand that human behaviours are integral to the heart of organisations and ergo, people in that system need support, guidance and nurturing. This is how we affect the deepest part of the culture or personality of the organisation, by positively stimulating people's individual character, so the reach is systemwide.

Very few organisations understand where to start when embarking on a change initiative. The focus on *business agility* is valuable yet all too often falls short on simple steps that concentrate on process change or leadership-centric initiatives. While useful, coaches often miss the essential aspect of any organisation to change successfully - and that is its people.

"True wisdom comes to each of us when we realise how little we understand about life, ourselves, and the world around us."

A focus on human behaviour allows us to observe and facilitate relationships in our organisation. Coaching is a means to get people to reflect heuristically on themselves and their impact on the world around them. Such focus often creates an epiphany, a self-realisation moment that in turn will move them through a threshold of change. To watch this is magical and immensely satisfying. The metamorphosis that takes place internally and between people has the potential to last a long time because it goes deep. You often won't recall a coaching conversation, but you will certainly remember how a coach made you feel.

To summarise, I would distil my reflections into these points:

- Look beyond process and go deeper into organisational principles (which start with people). *The ten principles of enterprise change,* of which human behaviours are part, very much go into the more profound organisational principles.
- Human behaviours and models such as Leary's circumplex can be applied in any organisational change initiative. Context is irrelevant.

- People make the culture and experience it as the personality or mood of the organisation. As coaches and consultants, we, at best, focus on adjusting the climate or the organisation and leave, but then things habitually go back to the way they were. Instead, we need to do more than this to guide people through change and help them and ergo the organisation to flourish.

Focusing on human behaviours allows us as coaches to explore the deepest levels of organisational change. This is important. We must seek to learn more about ourselves before we can embark on a journey to help others. As Socrates put it, "True wisdom comes to each of us when we realise how little we understand about life, ourselves, and the world around us." As coaches, reflecting on human behaviours will help us identify how best to guide and leave a positive impact on the people and the wider organisation for meaningful change and ultimately a sustainable transformation.

This article was originally published at
https://baa.tco.ac/34oD

About Zia Malik

For **Zia Malik**, organisational change doesn't just mean adopting new processes and working in new ways. It isn't even simply about thinking in new ways. True transformation requires challenging beliefs, and that means changing people from the inside.

Zia is one of a small number of UK Certified Enterprise Coaches and professionally trained coach in business and personal coaching, accredited by the International Coach Federation. He has worked as an enterprise coach in a range of industries. Read his Zia's bio at beliminal.com.

Ways to Contact Zia

beliminal.com
zia@beliminal.com
www.linkedin.com/in/ziamalik
twitter.com/mrziamalik

Our Editors

The Masterminds

About Cherie Silas

Cherie Silas, MCC is Managing Director of Tandem Coaching Partners and the Director of Education at Tandem Coaching Academy.

She is a Certified Enterprise Coach (CEC), Certified Team Coach (CTC), and Human Systems Coach (CHS).

Cherie also serves as a professional coaching instructor and mentor coach helping coaches improve their core coaching competencies and skills. She is a Certified Supervision Practitioner (CSP) for experienced coaches helping them look at the work they are doing with their clients to strengthen those client relationships for more effective coaching engagements.

Tandem Coaching ACADEMY

Ways to Contact Cherie

tandemcoaching.academy

cherie@tcagile.com

www.linkedin.com/in/cheriesilas

twitter.com/CoachingTandem

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.

He is the co-editor of Agile Coaching: Wisdom from Practitioners and Why Agile Works: The Values Behind the Results.

Michael holds a PhD in Computer Science from MIT.

HEART HEALTHY **SCRUM**

Ways to Contact Michael

www.hearthealthyscrum.com

michael.delamaza@gmail.com

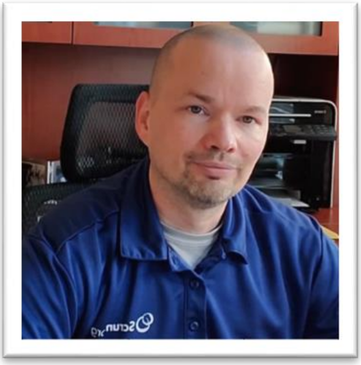
www.linkedin.com/in/michaeldelamaza

Our Volunteers

They Made This Publication Happen

Abhinav Roy

I am an Agile Coach based out of India. In addition, I am also a certified Yoga instructor and I am trying to blend flexibility of body with agility of mind to bring about more consistent results in my personal and professional life..



Alex Kudinov

Alex is a Technologist, Professional Coach, and Trainer. He delivers business competitive advantage to his customers by growing and nurturing great agile development organizations.

At Tandem Coaching Academy where we keep Agile non-denominational, Alex works with Agilists of all walks, bringing agile and professional coaching closer together. Alex holds the International Coach Federation Professional Certified Coach (PCC) accreditation.

Andrew Lin

Andrew has been practicing and teaching Scrum for about 15 years. He trains and coaches Scrum Masters, Product Owners, Teams, and Executives. He helps teams and organizations to deliver twice the value in half the time.

He hosts Meetups and volunteers in agile/coaching communities. Andrew is the first LST (Licensed Scrum Trainer) trained and certified by Jeff Sutherland Scrum Inc. in the Greater China area. He works and lives in VA and Taiwan. You can reach Andrew at andrew.lin.agile@gmail.com

Anu Smalley

Anu is the President and Founder of Capala Consulting Group, LLC where she specializes in Executive coaching and Agile Transformations.

She is also a Certified Scrum Trainer® and Certified Enterprise Coach® - providing education and coaching, consulting services to clients across the United States.

Chester Jackson

Chester Jackson enjoys the opportunities of teaching, mentoring, and coaching individuals and teams at all levels of an organization.

He is passionate about moving others through awareness into new action to help them improve the impact they have in their jobs, family, and the world.

Christine Thompson

Christine Thompson is an Agile Coach, a Team Coach and an Individual Coach. She is committed to helping you to become the best version of yourself.

As an ICF Associate Certified Coach, a Scrum Alliance Certified Team Coach, an NLP Master Coach and being ORSC Trained, she works with teams to promote an Agile mindset, and a set of behaviours resulting from that mindset, which allows them to focus on delivering value whilst being responsive to change.

Gary Hansen

Gary is a self-proclaimed "learning instigator." He has worked with agile approaches since 2009 and his major engagements have been with United Health Group, Target, 3M, Best Buy, Thrivent Financial, & TCF Bank.

He works with all levels of an organization to ensure that transformation efforts are aligned. Gary is a Scrum Alliance Certified Team Coach (CTC), International Coach Federation Associate Certified Coach (ACC), and has 13 additional certifications. Gary has an M.Ed. in HR Development.

Gitta Toussi

Gitta is an Enterprise Lean Agile Coach, using Agile practices for Digital Transformation in diverse industries. Her specialty is in practical and lean technology implementation with focus on business agility and market competitiveness.

She is equally at home with various delivery methodologies, and her passion as a servant leader is to bring out the best in people and their talent. Gitta earned her Agile certifications from Chicago State University, PMI, Scaled Agile Inc., and Scrum Alliance.

James Black

James has spent the past 15 years in software development in various roles from Software Developer, Project Manager, and now serving his organization in the role of an Agile Coach & Scrum Master.

His goal is to help organizations understand different frameworks and processes that can lead to successful outcomes. James believes that by encouraging teams to continue learning, challenging our current processes and empowering people to make a difference is how we succeed and deliver true value.

Jayaram Hegde

Agile Coach | Consultant | EI Practitioner | Associate Certified Coach from International Coaching Federation | Transformation Consultant & change Enabler with 13+ years of Software industry experience. Carry rich experience working across organization as Coach supporting team, leadership and business.

Jody Bowen

I have been employed at a Fortune 500 company for 27 years where I am a Scrum Master and Agile Coach. I enjoy teaching what I know and have learned to empower others to be there best.

I am currently our pursuing my CTC. I enjoy spending time with my family and friends and look forward to traveling again soon.

Joseph Jones

Joseph Jones is a Professional Coach, servant leader and change agent. He's an advocate of the growth mindset, practicing agile approaches. He's active in the agile/coaching community and volunteers on the Agile Alliance's Diversity and Inclusion committee.

Joseph is a member of several meetup groups such as Agile in Color, Tandem Coaching Academy, and Agile Coaching Circles to name a few. He holds a MFA from The George Washington University in Visual Communication.

Lance Zhang

Lance is Scrum Alliance Certified Agile Coach (CTC) and Agile Leadership Journey Guide. He has multiple years' experience in leading agile transformation and coaching individuals, teams and organizations.

He supports global companies and startups in various industries. Lance is a community leader and frequent conference speaker in China. He is keeping promoting agile mindset, and supporting individuals and organizations on their journey to agility.

Manohar Jakkireddy

Manohar is working as Agile Coach. Expert in Organizational Transformation, passionately helping organizations to embrace Agile and continuously improve their Agility.

Manohar practices include coaching, mentoring and training in varied agile frameworks including Scrum, Kanban, XP and other Scaling frameworks.

Martin Lambert

Martin is an independent agile practitioner, scrum master, consultant, department level agile coach, trainer and delivery-lead. Exiled from the north / living in the south of the UK, usual clients are big corporations for a year or so at a time, engaged as some form of embedded team coach.

Specialities are spinning up a new scrum team, guiding them and their managers in lean agile thinking with context senitive pragmatism. When not doing that, he's riding a bike up a hill. For pleasure.

Matt Kirilov

Matt Kirilov has a well-rounded set of skills that deepen in both the coaching, teaming (team development), and group psychodynamics competencies. He is passionate about leveraging relationships to partner with clients striving for positive change in organizational systems.

Professionalizing Agile Coaching and adopting core coaching principles by the Agile community are very close to Matt's heart. The interdisciplinary intersections are the places where you can find him working on his growth edge.

Paulo Dias

Paulo is a Professional Certified Coach (PCC) by the ICF on the journey to become a CTC/CEC by the Scrum Alliance and a ICE-EC by ICAgile. He has 25 years experience in software development.

In the last 12 years, Paulo has been working on a variety of different roles as Enterprise and Business Agility Coach, Consultant, Trainer and Mentor in many different industries including Airlines, Telecommunications, Financial Services, Insurance, Automobile and Energy.

Rohit RatanMani

Rohit is an Agile Coach, helping individuals, teams and leaders to develop a growth mindset to be top achievers in their respective work area and in personal life (to an extent). He has worked on multiples large and complex Agile Transformations with different consulting, financial and product based organisations.

He is a Certified Team Coach (CTC) from ScrumAlliance, Associate Certified Coach from ICF and an ICAgile Trainer for Agile Coaching Track.

Steve Matthew

Steve has over a decade of professional experience and has worked in various roles, such as Business Analyst, Scrum Master, and Agile Coach.

He has a broad array of real-world experience and has worked in a variety of industries, such as Information Technology, Retail, Banking, and Insurance. He is an Accredited Kanban Trainer (AKT) and SAFe Program Consultant (SPC). Steve enjoys writing articles and public speaking so that he can help bring awareness within the Agile community on commonly misunderstood or difficult Agile topics.

Ted Wallace

Ted Wallace is currently an Agile Coach at Principal Financial Group. He has completed Master of Science degrees in Computer Science and Physiology and is a certified ScrumMaster Professional (CSM, CSPO, CSP, CTC) and a registered corporate coach (RCC) with thousands hours of coaching sessions completed.

His talented wife, Danielle, is from the Netherlands. They have three amazing children, Jace, Kyran, and Myka He is an author of "Total Brain Coaching" and "The Coherence Code".

Tandem Coaching
ACADEMY

HEART HEALTHY **SCRUM**