

Agile Coaching

Wisdom from Practitioners

Version 1.0

Editors

Michael de la Maza, CEC
Dhaval Panchal, CEC, CST

Contributors

Brock Argue, CEC
Roger Brown, CEC, CST
Michael de la Maza, CEC
Bob Galen, CEC
Gene Gendel, CEC
Daniel Gullo, CEC, CST
M. Kelley Harris, CST
Jon Jorgensen, CTC
Mark Levison, CST
Lizzy Morris, CST
Simon Orrell, CEC
Dhaval Panchal, CEC, CST
Cherie Silas, CEC
Chris Steele, CEC
Mark Summers, CEC, CST
Chris Waggoner, CEC

Agile Coaching

Wisdom from Practitioners

Version 1.0

Agile Coaching

Wisdom from Practitioners

Version 1.0

Editors:

Michael de la Maza, CEC & Dhaval Panchal, CEC, CST

Contributors:

Brock Argue, CEC
Roger Brown, CEC, CST
Bob Galen, CEC
Gene Gendel, CEC
Daniel Gullo, CEC, CST
M. Kelley Harris, CST

Jon Jorgensen, CTC
Mark Levison, CST
Michael de la Maza, CEC
Lizzy Morris, CST
Simon Orrell, CEC
Dhaval Panchal, CEC, CST

Cherie Silas, CEC
Chris Steele, CEC
Mark Summers, CEC, CST
Chris Waggoner, CEC

AGILE COACHING: WISDOM FROM PRACTITIONERS

Edited By: Michael de la Maza, CEC & Dhaval Panchal, CEC, CST

Contributors: Brock Argue, CEC; Roger Brown, CEC, CST; Michael de la Maza, CEC; Bob Galen, CEC; Gene Gendel, CEC; Daniel Gullo, CEC, CST; M. Kelley Harris, CST; Jon Jorgensen, CTC; Mark Levison, CST; Lizzy Morris, CST; Simon Orrell, CEC; Dhaval Panchal, CEC, CST; Cherie Silas, CEC; Chris Steele, CEC; Mark Summers, CEC, CST; Chris Waggoner, CEC

Copyright ©2017 by Michael de la Maza
All rights reserved.
Printed in the United States of America.

The text of this book is set in Adobe Garamond & Hypatia Sans Pro
Book layout and cover design by THDesign, Inc.

First Edition: August 2017

Table of Contents

Contributors

Brock Argue, CEC

Biography.....	Page 7
The Coaching Mindset	Pages 8–9

Roger Brown, CEC, CST

Biography.....	Page 10
An Example of a 5-Minute Coaching Session with Roger Brown	Pages 11–12
The Agile Coaching Mindset: An Interview with Roger Brown	Pages 13–14

Michael de la Maza, CEC

Biography.....	Page 15
Coaching or Mentoring? A One Act Play	Page 16
The Existential Dilemma of the Agile Coach.....	Page 17
How Much Money Should I Save?	Pages 18–19
Stress: Management vs. Team	Page 20
Target State Change vs. Evolutionary Change	Page 21
SAFe.....	Page 22

Bob Galen, CEC

Biography.....	Page 23
Agile Coaching – An Awful Truth	Pages 24–26
We’re Coaching the Wrong People!.....	Pages 27–30
Is It Worth the Energy?	Pages 31–33
The Agile Coaching Dilemma.....	Pages 34–36
What the World Needs is More Prescriptive Agile Coaches	Pages 37–40

Gene Gendel, CEC

Biography.....	Page 41
Unspoken Agile Topics.....	Pages 42–61
Bad Choice of Verbs Associated with “Agile” by ELF People.....	Pages 62–63

Daniel Gullo, CEC, CST

Biography.....	Pages 64–65
5 Years Ago	Pages 66–69
Make Agile Great Again	Pages 70–72
The Lost Art of Management.....	Pages 73–75

M. Kelley Harris, CST

Biography.....	Pages 77
Innovation Trends to Befriend: Empathy, Safety, Culture and Invitation	Pages 78–79

Jon Jorgensen, CTC

Biography.....	Page 80
16 Questions for Setting-Up a Coaching Alliance	Pages 81–82

Table of Contents continued

Mark Levison, CST

Biography.....	Page 83
“Because Our Competitors Are” is No Reason to Become an Agile Organizaion	Pages 84–86
Simplicity.....	Pages 87–88

Lizzy Morris, CST

Value Adaptations Even When It’s “Ouchy”	Page 90
--	---------

Simon Orrell, CEC

Biography.....	Page 91
Executing Large EPC/EPCM Projects Using Scrum Values and Principles.....	Pages 92–109

Dhaval Panchal, CEC, CST

Biography.....	Page 110
My Favorite Coaching Experience	Pages 111–113
The Coaching Mindset	Pages 114–116

Cherie Silas, CEC

Biography.....	Page 117
The Agile Manifesto — What It Means to Me	Pages 118–119
Is More Really Better?.....	Pages 120–122
Tips From the Trenches	Pages 123–125
What Does Sustainable Pace Really Mean?	Pages 126–127

Chris Steele, CEC

Biography.....	Page 128
The Successful Coach.....	Pages 129–131

Mark Summers, CEC, CST

Biography.....	Page 132
How Do I Use My Agile Coach?	Pages 133–135

Chris Waggoner, CEC

Biography.....	Page 136
When It’s No Longer Relative	Pages 137–139

Brock Argue, CEC



Brock Argue specializes in taking an organizational approach to Agility, recognizing that all aspects of the business benefit from the application of Agile values and principles. Drawing from many different disciplines, his style of facilitation creates an environment in which high-performing organizations can emerge. He is focused on improving organizational Agility through cultural change and is dedicated to growing his connections within the Agile community.

Brock's first exposure to Scrum and Agile methods occurred in 2006 when he and two of his peers introduced Scrum into the start-up company they were working for at the time. Convinced that there must be a better way to work that released the team from repeating past mistakes, Brock helped his team run an experiment to apply Agile principles and practices to their work. The success of that experiment led to Scrum being rolled out across the organization.

As a Certified Enterprise Coach (CEC) through the Scrum Alliance®, Brock's previous work includes agile transformations at Digital Oilfield and ADP. Based in Calgary, AB, he is currently the Agile Coach at Benevity, Inc., a world-class software social enterprise that's helping change corporate philanthropy while accomplishing its social mission.

In addition to his work coaching organizations, Brock provides coaching to individuals seeking to grow in their Agility as the co-founder and coach of Superheroes Academy (<https://superheroes.academy>).

The Coaching Mindset

Original Source: <https://www.superheroes.academy/blog/2017/3/9/coaching-mindset>

What is a Coaching Mindset?

As coaches, we're told that it's important to have a coaching mindset. Being able to demonstrate this thought pattern is often a requirement of coaching certifications and is something we look for when reviewing applications for the Scrum Alliance Certified Team Coach (CTC) and Certified Enterprise Coach (CEC) certifications. As such, this is a main focus of our mentoring program here at Superheroes Academy. Now, the question remains, what does it mean to have a coaching mindset? Let's explore this question together.

Characteristics of a Coaching Mindset

When it comes to developing a coaching mindset we often think about what we do as coaches. This includes listening, asking powerful questions, holding space and facilitating structured conversations. While doing is important, these are techniques we use in our coaching and are not who we are as coaches. Our mindset becomes visible in what we do and how we approach coaching situations, however, our mindset exists more in who we are as coaches.

Who Are You as a Coach?

Although all coaches are different and have a different style and approach to coaching, there are some character traits and attributes that good coaches have in common. Good coaches are:

- **Emotionally Intelligent**

Popularized by Daniel Goleman in his book, *Emotional Intelligence: Why It Can Matter More Than IQ*, emotional intelligence (or EQ) involves growing oneself in four areas: self-awareness, self-management, social awareness and relationship management. This journey starts with self-awareness as an individual becomes more able to know and understand their thoughts before they act. Emotionally intelligent coaches possess a greater degree of empathy towards others, including their clients.

- **Present**

"Be here, prepared to be nowhere else," says Susan Scott in her book, *Fierce Conversations: Achieving Success at Work & in Life, One Conversation at a Time*.

- **Curious**

My favorite book when I was young was *Curious George* by H. A. Rey. Each story in the series started with the statement that, "This is George. He was a good little monkey and always very curious." Good coaches possess a genuine curiosity and desire for continuous learning.

- **Courageous**

As people, coaches are not free from fear and doubt, however we overcome this for our client's sake. As Nelson Mandela put it, “. . . courage is not the absence of fear, but the triumph over it.”

- **Neutral**

Maintaining a neutral stance is a key to the coaching relationship. Our role is not to judge, but rather to work with the client to discover new possibilities and help them reach their full potential.

- **Accountable**

Good coaches hold their clients accountable for the plans and decisions they make. Coaches also maintain their own accountability for the process of coaching and upholding many of the mindsets discussed above.

- **Action-oriented**

In the recent book, *The Founder's Mentality: How to Overcome the Predictable Crises of Growth* one of the main attributes described by Chris Zook and James Allen as critical for success is having a “bias toward action.” Coaches should have this disposition too. Coaching conversations are not meandering Sunday drives in the wilderness, but rather a trip intentionally guided towards a destination, which leads to focused action on behalf of the client.

- **Sold on their client's abilities**

Our belief as coaches is that our clients have all they need to succeed — they just may not realize this yet. Coaches help their clients discover what they already know and have a firm belief in their ability to reach their potential, even when the client waivers. As a coach, “my certainty is greater than your doubt.” – Susan Johnston (*Coach Skills for the Agile Workplace* training course, Calgary March 2016).

Who are you as a coach? Which parts of the coaching mindset resonate with you the most?

Roger Brown, CEC, CST



Roger Brown is an independent Agile Coach in San Diego who provides training, consulting & coaching services in Scrum and enterprise Agile adoption. He is a veteran software engineer and team lead with experience at the U.S. Veterans' Administration, Dartmouth College, Vicinity Corporation, Microsoft and TeleAtlas. His coaching and training clients include over 70 companies from small startups to multinational banks.

Roger is a Scrum Alliance™ Certified Enterprise Coach and Trainer. He is a reviewer and founding member of the Certified Scrum Coach, Certified Enterprise Coach and Certified Team Coach Programs at the Scrum Alliance. He has facilitated open coaching clinics at several Agile conferences since 2009.

An Example of a 5-minute Coaching Session with Roger Brown

Background: Roger Brown (coach) sat down with Michael de la Maza (client) for a five minute coaching session. Afterwards, they analyzed Roger's approach and its impact on Michael.

5-Minute Coaching Session

Roger: Tell me a little bit about your environment? Tell me about yourself at work?

Michael: Excellent. The situation I'm having and it's a situation at home, not at work. It's my girlfriend is interested in switching jobs and she's recently asked me for help in switching the job. I have like a big stake in this. I have an opinion, but I don't really want her to switch jobs and so I'm not too sure I can support her in switching jobs.

Roger: Have you said that to her straight up?

Michael: No. I haven't done that to her straight up.

Roger: All right. Kind of side bar, what I'm going to do is given that story, I'm going to reflect it back to you to see if we're on the same page. What I hear is you have a situation at home. I'm going to make the assumption that there's some tension around this, at least for you, if not both of you. There's a big decision to be made that has impact to both of you, possibly the relationship and you're being asked to help and I'm not sure if you've been asked, what your feelings are about it, so maybe I'm going to ask about that. You said she asked you for help, I don't know what the nature of that help would be, so my first question is, has she asked you what you think about it in terms of how would it impact the relationship, which is kind of the system here, right, the third party, from a systemic coaching standpoint is there's me, there's you, and there's us. That's 3 parties. Has she asked you how you feel about it?

Michael: She has not asked me how I feel about it and I haven't offered my thoughts on how I feel about it.

Roger: She has asked for help in what way, help thinking through it?

Michael: Yeah, help thinking through it, what's her next step, how should she explore, how should she find out what she wants to do next, things like that.

Roger: I see. It isn't a specific job offer. It's a change, a desire for a change?

Michael: Right.

Roger: Is it a big step change in terms of different career or different geography?

Michael: I don't think it will be a big step in either one of those, but it might be a big step financially.

Roger: Is that something you've discussed?

Michael: Only a little bit and what she said, "I'm concerned about making a lot of money."

Roger: That was her statement or your statement?

Michael: That was her statement and that's also my main concern.

Roger: How far have you gotten into a helping activity so far?

Michael: We just started. We have a meeting scheduled for Sunday morning.

Roger: Are meetings a common activity in your relationship?

Michael: No. This is like the first meeting that we've had.

Roger: Sure, we'll stop. I will tell you, I don't know if this will make you feel better or worse, but this is my 40th year of marriage, 42nd year with my wife and when we have something that one of us is avoiding, we have a meeting.

Michael: Okay, so it's a practice at even very high levels of experience?

Roger: Absolutely. It doesn't even have to be avoiding because it's unpleasant. It might be just because oh, I don't want to be bothered with that right now. I've got other things I want to do.

Michael: Thanks a lot.

Roger: We do that, because those things are going to continue to come up, things that require joint focus essentially will continue to happen in a relationship that is sustained.

Michael: Excellent.

Analysis and Feedback

Roger: I didn't really get to any coaching. I just asked a lot of questions trying to understand this circumstance.

Michael: Really, it was super helpful to me. Your questions encouraged me sort of to view what was going on from a higher level of abstraction, so it took me from sort of being in it, to being an observer of it and looking at the dynamics of it.

Roger: Fabulous.

Michael: Yeah.

Roger: Fabulous and maybe it's because you're a coach or maybe just because you're super smart. If I had continued, that was where I was trying to get, but you already got there.

The Agile Coaching Mindset: An Interview with Roger Brown

Michael: What is the Agile coaching mindset?

Roger: The Agile Coaching Mindset, hm, let me think about this. Coaching mindset in general is a thing that one learns as one becomes a coach. The Agile variation is just, I guess, around domain knowledge of Agile practices, principles and common challenges.

Coaching mindset has to do with serving people by helping them solve their own problems, so that mindset is about being open to possibilities, open to the likelihood that you, as the coach, don't know the solution to whatever problem is being addressed and not necessarily agreeing that whatever problem is expressed by the client is the actual problem.

It has to do with the ability to help the coaching client or the coachee discover more clarity around what they're trying to achieve and some ways to achieve it. In the Agile world, I think we would add to that the idea of incrementalism — where we don't necessarily try to solve a problem in one step, but we make steps towards a solution as we're working.

Michael: If an Agile coach is in the coaching stance and the client says, “I just want you to solve the problem” what might the Agile coach say?

Roger: There are a lot of things a coach can say and before a person says that, the coach is going to be ready to have that be a possibility. I'm going to just give you an example, when we do our coaches clinics at the conferences and I engage a client knowing I have a very short time to work with them, the first thing I do is ask a series of questions to determine if this person wants coaching or consulting. Coaching is what I just described and consulting is just telling them information or giving a possible solution to the stated problem.

The questions I ask are around the nature of what you're looking for, but also trying to discern if that problem or challenge, whatever it is, is clear in the client's mind. As a coach, do you see that there might be some further layers to it? You would want to be thinking about what questions you would ask to determine those things and if it becomes a coaching circumstance, try to peel back those layers.

In the coaches clinic example, I always start with: Is this person asking about something that they might possibly be able to determine on their own or are they simply asking for information that they don't know. That helps me decide if my job is to give them a brain dump of what I know in the domain or is it to give them true coaching to help them solve a problem that they have been thinking about or working with for some time that they might have the solu-

tion to themselves — keeping in mind that I may not have the same solution in mind.

If someone says “I just want information” then I start there and watch for a greater context, I just had this happen, by the way. Someone asked me to do some coaching with them and then someone else said that person doesn't really want coaching. She wants consulting. She is in a hurry and wants answers, because she really doesn't know enough yet to know what to ask.

My approach when a person says “I just want information” is to ask a few questions around “Are you sure?”. Those questions have to do with the problem are they trying to solve or the improvement they are trying to make. The answers will reveal to me, from my experience, if there might be layers of inquiry or discovery around that topic.

Michael: Is there ever any situation where you're not sure whether or not the client is able to solve the problem or is able to work through the problem and how do you deal with that? How do you self-manage for those situations?

Roger: At the limit, there is a circumstance where, as a coach, you need to have the courage to say “I don't think I can help you”. That's the extreme, where you say I'm not the right person for you, either because I don't know what I need to know to help you or because I don't think, from our conversations, that you can solve this problem for some reason. There is a whole spectrum of possible reasons. The most common one is that the client doesn't really want to solve the problem. This is the one that we most commonly encounter, because we're talking about making changes in people's behaviors and companies' cultures and such.

As a coach you may be able to detect this circumstance because you have been there, done that, or seen something like it. If you don't have that expertise, then you have to work with the client long enough to see if there is a solution. That can take a while. I don't know that I answered your question though.

Michael: Absolutely. When you are interviewing a coach or when you're, say, evaluating an application for a coaching certification, how do you detect or identify the coaching mindset in someone else?

Roger: That's pretty easy. You can get it from language. Someone who's in a consulting mindset is using words like “advise”, “tell”, “suggest”, “guide them to”. Someone who is a coach uses language about asking and exploring and considering things that are not necessarily obvious. So the language of a coaching mindset is more asking than telling.

Back to your original question, the Agile coaching mindset, as we have come to learn over the years, is a balance between asking and telling. There are times when you can go too far one way or the other. Most “Agile Consultants” go pretty far in the telling. I've seen some cases where “Agile Coaches” go too far in the asking - where it's not satisfying to the client to continually be answering questions and never landing anywhere. That's how I detect the presence of a coaching mindset, from the language of the discussion or the written word in the case of an application for certification (CTC or CEC).

Michael de la Maza, CEC



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

Michael is a Co-Active Coach and is co-organizer of the BayALN, the agile user group in San Francisco and the organizer of the BayALN Certified Coach Special Interest Group (pathtoctc.org). He loves playing, creating, and sharing games and co-organized the 2010 Agile Games Conference, the 2011 Agile Games Conference, and the 2016 Agile Games West Conference.

He serves on the Scrum Alliance's Certified Team Coach (CTC) review committee and the Virtual Coaching committee. He enjoys mentoring Scrum Masters and Agile coaches who want to deepen their understanding of Agile. He also mentors and invests in startups through Techstars Boston.

He can be reached via email at michael.delamaza@gmail.com and on Twitter @hearthealthyscr.

Coaching or Mentoring?

A One Act Play

Scene: Coachee and coach are in a small, carpeted study room at the company's office shortly after noon on Monday. Both have just returned from a disastrous, humiliating meeting with executives in which they were both belittled and criticized. The coachee is slurping Starbucks and looking around with a harried, frenzied expression. The coach is clean shaven and is wearing jeans and glasses.

(Coachee, lugubriously)

“My problem is that I'm not good at anything. That's why I'm constantly running into trouble. I have nothing to offer.”

(Coach, helpfully and confidently)

“You have something to offer everyone.”

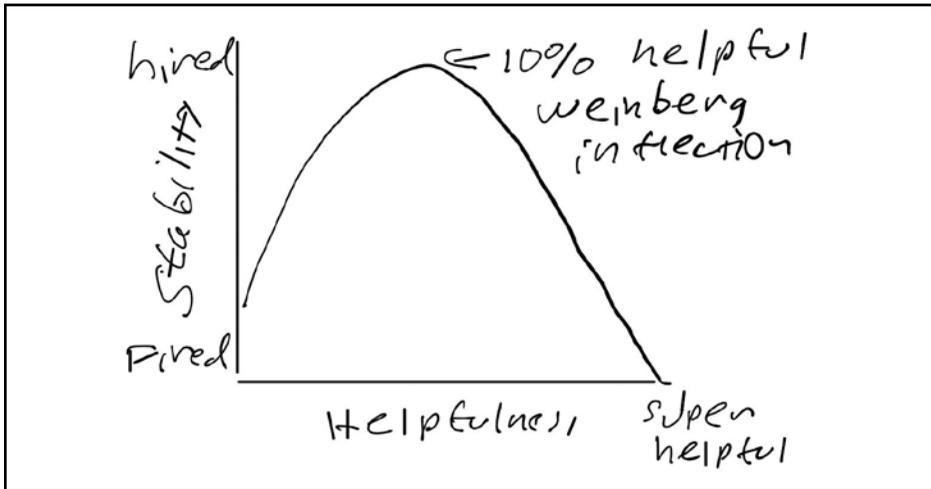
(Coachee, sarcastically and loudly)

“Really? What do I have to offer Bill Gates??”

(Coach, realistically)

“You know more about failure than he does.”

The Existential Dilemma of the Agile Coach



“Never promise to solve a problem. Promise 10% improvement to a working system at best. A doubling of productivity/throughput makes you look good, at someone else’s expense. That person then becomes your enemy. Not worth it.”– Gerry Weinberg, *Secrets of Consulting*

How Much Money Should I Save?

How much should I, an independent Agile coach save, to ensure financial stability?

This question is important because Agile coaches get unexpectedly terminated far more frequently than other consultants. My friend Brian Wills likes to tell the story of the 18 month engagement with a Fortune 50 company that was so compelling he moved his entire family 2,000 miles only for the engagement to end after six months. This prompted him to create Wills Rule of Agile Coaching: “No matter what you are told, assume that the engagement will end within 6 months.”

Here are some examples of my unexpected terminations:

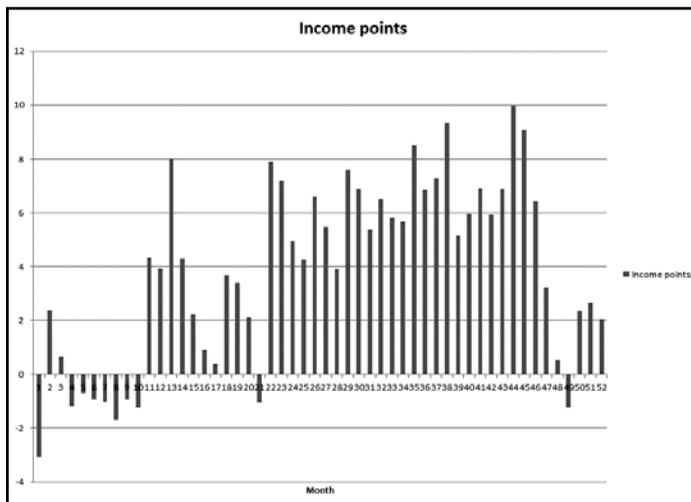
- After six weeks in a three month engagement, I was asked to not return.
- After nine days in a multi-month engagement, I was told that the engagement was over.
- After one month in a multi-month engagement, I was fired while boarding a plane to the client.

In none of these cases did I receive any explicit feedback prior to the termination.

While quality statistics are hard to come by, all of the evidence I have is that Agile coaches find that their engagements end prematurely far more frequently than, say, software consultants or management consultants do.

That means the standard emergency fund advice — have 3 to 6 months of savings on hand — may not be sufficient to support me.

How many months of savings should be in my emergency fund to ensure that I have no



more than a 1% chance of running out of money at any time in the next 10 years?

Let's start with some raw data. Here are my net earnings (in 'income points') for the last 52 months:

Here are a few facts that immediately jump out at me:

Between month 1 and month 10 my income was -7.7 points so I need at least that much money in my emergency savings.

My income is auto-correlated: When the going gets bad, it is really bad (months 4-10) and when the going is good, it is really good.

A good month covers several bad months.

My income is not normally distributed.

Because the results are so highly auto-correlated, a simple Monte Carlo is not sufficient to provide a good estimate of how many income units should be in my emergency fund.

Here is an approximate method that will generate a time series with the same average with a similar autocorrelation as the original time series:

1. Randomly select a starting month (1-52).
2. Randomly select a number of months (from 1-10) from the starting month (not that month 52 wraps around to month 1).
3. Repeat steps 1 & 2 until 120 months (10 years) have been selected.

For example, let's say that the first starting month (step 1) is 17 and the randomly selected number of months (step 2) is 5. Then months 17-21 will be the first five months of our 240 month series.

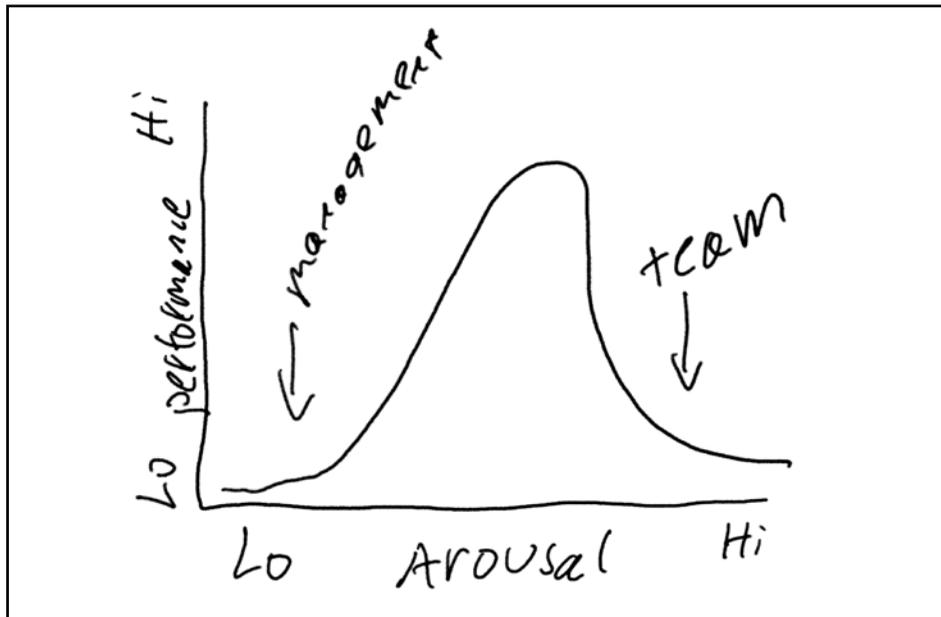
Informally, it's clear that month 4 is the worst starting month and the worst sequence is months 4-10. However, the likelihood of going through this exact sequence twice in a row is extremely small. So I would expect the emergency fund to be between 7.7 (months 1-10) and 15.4 (twice months 4-10). (By sheer coincidence, both the sum of months 1-10 and months 4-10 is 7.7).

The answer turns out to be 9.9 income points. In order to keep the chances that I will go bust to less than 1% over the next ten years, I need to have 9.9 income points in my emergency fund. Of course, if I had just this amount my emergency fund would be zero in this worst case scenario. So as a practical matter, I plan to keep 11 income points in my fund.

I'm glad I did this analysis because this is approximately three times the amount in my current fund. Also, my monthly expenses are approximately one income point so I need 11 months of expenses — far higher than the 6 month emergency fund recommended for the typical consultant.

Stress: Management vs. Team

“The **Yerkes–Dodson law** is an empirical relationship between arousal and performance, originally developed by psychologists Robert M. **Yerkes** and John Dillingham **Dodson** in 1908. The **law** dictates that performance increases with physiological or mental arousal, but only up to a point.” – Wikipedia



Target State Change vs. Evolutionary Change

When an organization undergoes an Agile transformation what transforms?

One of the most fundamental transformations is in the organization's understanding of how change and improvement take place. The company transitions from a target stage change approach to an evolutionary change approach.

Target state change involves three steps:

Assess -> Design -> Implement

Target state change is the bread and butter of management consultants and hero managers: the consultant or manager assesses the current system of work, designs a better system of work, and then the company implements the new system of work. When a management consultant says they are helping a company with Agile, what they almost certainly mean is that the Design step is informed by practices that were developed or popularized by the Agile community.

Evolutionary change is radically different:

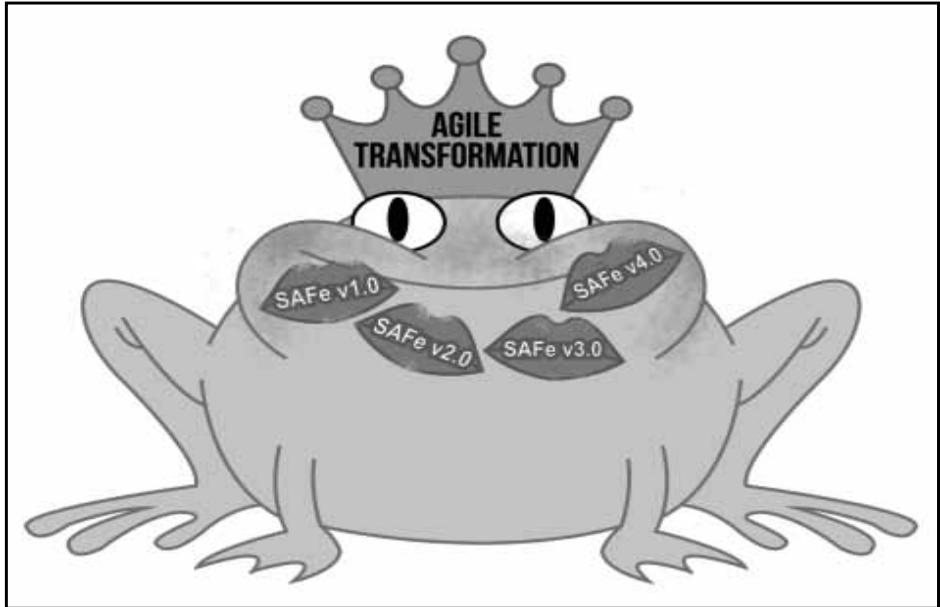
Experiment -> Inspect -> Adapt

In evolutionary change, the system is changed, the new system is evaluated, and if the change results in improvement, it is kept. If it does not, it is rejected. What this means is that Agile looks different at every company.

An almost foolproof sign that a company has a target state change approach to Agile is that they copy and paste a large number of low-level practices from highly Agile companies. Spotify, with its evocative squad/guild/tribe/chapter organizational design language, is one of the companies most often copied. This despite the fact that Spotify itself has pointed out that copying the 'Spotify model' is ill advised (source: <https://www.infoq.com/news/2016/10/no-spotify-model>). (On a snarky note, if management consultants really wanted to copy and paste Spotify's approach to management, they should share with their clients that Spotify ran for seven years without a single MBA on their leadership team.)

Misunderstanding Agile transformation as a target state change to a system of work chocka-block with generic Agile practices is one of the biggest mistakes (or greatest opportunities) in the world of Agile coaching.

SAFe



Bob Galen, CEC



Bob Galen, is an Agile Methodologist, Practitioner and 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 methodologies and practices. He is Director, Agile Practices at Zenergy Technologies, a leading Agile transformation company. He is also President and Head Coach at RGCG.

Bob regularly speaks at international conferences and professional groups on topics related to software development, project management, software testing and team leadership. He is a Certified Enterprise Coach (CEC), Certified Scrum Product Owner (CSPO), and an active member of the Agile & Scrum Alliances.

He's published three Agile-focused books: *The Three Pillars of Agile Quality and Testing* in 2015, *Scrum Product Ownership* in 2009 (2nd Edition in 2013), and *Agile Reflections* in 2012. He's also a prolific writer and blogger (www.rgalen.com) and podcaster (www.meta-cast.com)

Bob may be reached directly at: bob@rgalen.com or networking via: <http://www.linkedin.com/in/bobgalen>

Agile Coaching – An Awful Truth

Original Source: <http://rgalen.com/agile-training-news/2017/2/7/agile-coaching-an-awful-truth>

I attended an impromptu Agile coaches gathering about a year or more ago. It was a “coaching the coaches” session and it was very valuable. But an aspect of it has stuck with me ever since. One that I’ve mulled over and over and would like to share.

There were a group of coaches in attendance from the same client engagement, a large, multi-billion-dollar organization that had been going Agile for a couple of years.

When they decided to go Agile, one of the first things the client did was reach out to an Agile coaching firm for help. On the surface, that sounds like a good thing to do. However, the firm was largely staff augmentation focused, so that was their background and comfort zone.

They reacted like they would for any similar engagement. They recruited 10 disparate agile coaches, minimally vetted their experience, and aggressively negotiated their rates. Then they negotiated a global agreement with the client and on-boarded the coaches.

There was no engagement strategy nor much consistency across the various coaching approaches. There was also no coaching team. Instead, there was simply a group of coaches thrown into a very lucrative situation. And as coaches are wont to do, they started coaching . . .

Rates

Let’s take a diversion to approximate the cost of this endeavor. While I’m not privy to the exact rates, I know the ballpark. Each coach was probably signed up for ~\$1,200 / day while the client charge rate was ~\$2,500 / day.

The run-rate for each coach was ~\$625,000 annually. For ~10 coaches, the firm was paying ~\$6M per year. For a 2-year engagement, the total cost was approximately \$12M – \$15M, including coaching, certifications, and other training.

That sort of money should inspire and create phenomenal results, right?

Teams

The client quickly ramped from zero Scrum teams to about 150 Scrum teams. So, the coaches played a significant part in quickly scaling up the organization’s teams.

Their primary focus was downward to the teams. If you measured their success by how many teams were spun up and how quickly that was done, then they were quite successful.

Ultimate coaching costs per team were ~\$100,000.

Back to the Coaches

But let's get back to the clients' coaches in our meeting. To a person, they were sad.

It seemed while they were largely successful in getting teams on-board with Agile, they realized it wasn't enough to transform the organization.

They learned (and many had known before they joined) that you can't transform an organization at a team-only level, that any solid transformation needed the full engagement and participation of management and leadership.

Haunted

Part of the sadness at the meeting was the coaches were approaching the end of their engagement. The client organization felt that their value proposition had declined and the initial goal of achieving agile had been accomplished.

But the coaches knew differently. While the teams had been assimilated, the organization's leadership style remained the same. And the overall pre-agile culture remained the same.

In other words, the Agile teams were largely alone in their environment with no amount of leadership, management, or true cultural support. The coaches knew that the teams fledgling efforts would eventually revert to their previous approaches, that they would not stand the test of time.

Being professional coaches, they were quite sad about their efforts not resulting in sustainable change. They seemed to be wracked by questions like:

- Why wasn't there an overarching coaching strategy at the beginning?
- Why weren't we hired as and formed into a team for the engagement?
- Why wasn't there more of an on-site coaching leadership presence?
- Why didn't we challenge management and leadership more to engage and be a part of the transformation?
- Why didn't we intervene when the organization clearly misunderstood the nature of an agile transformation?
- Why did we continue to coach aggressively downward, when we knew that upward was the better direction?

And most daunting, why did we continue to coach when we knew we weren't making an impact in the best interest of the client's goals? Why didn't we leave instead of just cashing our checks and going through the motions?

And to be fair, it wasn't just the coaches who should have been asking these questions. Their firm should have been doing so as well. Especially since they were driving the overarching engagement strategy (or lack thereof) for this client's Agile transformation engagement.

In the End, A Tremendous Waste

The reason I brought up the funding model, was to show the incredible investment the client made in this effort. But it all seemed for naught.

In the end:

- The coaches felt like they had failed their Prime Directive, to coach an organizational-wide Agile transformation. And they did fail.
- The organization felt that they had done what was asked of them. They went Agile. But from an impact perspective, they all knew that very little in the way

of significant change (outcomes, performance, quality, culture) had changed. They had also failed.

- And they had spent \$15M in the process, for essentially another failed initiative.

From my perspective, this is an example of an incredible waste of effort, time, and funding. And it could have all been avoided with a much different strategy and approach.

Now I've joined the mood of those coaches. This entire tale makes me SAD! And what's even SADDER is this is not a unique outcome. This happens incredibly often in Agile transformations.

I've shared this tale so that you might avoid a similar outcome. Here are a few related posts that might be helpful to plot a different journey.

- <http://rgalen.com/agile-training-news/2014/6/9/agile-coaches-were-coaching-the-wrong-people>
- <http://rgalen.com/agile-training-news/2014/7/21/coaching-leadership>
- <http://rgalen.com/agile-training-news/2014/11/23/agile-coaches-trainers-have-you-walked-in-the-shoes-of-technical-management>

One where you, as an Agile coach, take a much more balanced and effective approach in your organizational coaching. Where you establish a leadership partnership early-on that trusts and engages your coaching at all levels of the organization. Where you spend more time “coaching UP” than you do “coaching DOWN”.

Or where and when this doesn't happen, you consider congruently moving onto greener coaching pastures.

Stay Agile my friends!

We're Coaching the Wrong People!

Original Source: <http://rgalen.com/agile-training-news/2014/6/9/agile-coaches-were-coaching-the-wrong-people>

SCENE 1: WHERE HAS ALL THE COACHING GONE?

I'm a Certified Scrum Coach (CEC) and I know quite a few CST's. Many of them offer training and coaching as part of their services. However, the typical client interaction, either with public classes or private training engagements, for many of them is as follows:

- Deliver a 2-day CSM class to a group of mostly client team members
- Rarely deliver a "talk to leadership" as part of the engagement, as theirs' is more of a team-centric play...

Then they move off on their merry way. One of the "tag lines" of the Scrum Alliance is "Transforming the world of work"; so many CST's get a sense of accomplishment at this point—feeling that the world of work has been, well . . . transformed.

This approach is training centric and coaching light to non-existent. It's also focused towards team members rather than management or leadership roles. It's my understanding that one driver for that is training is much more lucrative than coaching. Now I'm not saying that's the only driver, but I'd bet it's one of the primary drivers. It's also easier to "sell" training sessions and the related certifications over coaching.

My main issue with this approach is I'm not sure it sets the clients up for success. For example, I can't tell you how many times I've seen an organization send folks off to a CSM class and then assign them Scrum Master duties upon their return. These newly minted CSM's are ill equipped for the role of Scrum Master in the real world and they almost always fail in some way, which inevitably gets blamed on "Agile".

Or another pattern is that team members leave "hyped up" on the Agile principles and the promises of self-directed execution, go back home to their organizations, and then encounter the same dysfunctional management patterns without any tools to change how they engage with their leadership teams.

In both cases these CSMs need role models, examples, mentors, and coaching—in the trenches with their teams in order to be successful. It's also surprising how little of this is required to help them get over the hump and become more effective.

The other issue I have is that these folks seem to avoid traditional management in their training. Some even marginalize and/or somewhat demonize traditional management in the very companies they're training. They do this in the classes—painting a somewhat purist view towards Agile leadership that inevitably the company leadership falls short on.

But there is often little to no “reaching out” or “partnering” with the leadership folks in these organizations. And if coaching the teams themselves is minimal, then coaching leadership isn’t even attempted.

SCENE 2: WOW, I SPENT MOST OF MY COACHING TIME WITH “MANAGEMENT”

If you’ve followed my career in my writing, you are aware that I spent three years at iContact as a senior technology leader and the head Agile coach and evangelist. During my tenure, I was the primary coach and trainer for our teams in Agile methods and approaches. I taught Scrum and Kanban basics, Agile Requirements with User Stories, explored roles and responsibilities, and helped us scale with a modified Scrum of Scrums model, and even focused heavily on Extreme Programming practices.

I also coached our management team (team leads and functional managers) across UX, Quality & Testing, DevOps, Architecture, and Software Development. This went far beyond training and focused on situational leadership in moving their style and tactics from command-and-control to more servant leadership styles.

I joked at the time that I had two distinct jobs. I was the Director of our Technical teams reporting to our CTO. But I was also the organizational Agile Coach with responsibility for our overall transformation. Needless to say, I was fairly busy.

But here’s the thing, if you had asked me when I was working there, what percentage of time I spent coaching the “teams” vs. coaching “management”, I would have said 70:30. It just felt like I was doing way more team-based interaction and coaching.

But if you asked me the same question after I left the organization, I now flip the ratios around. I realize now that I spent a relatively small amount of my time at a team level. Instead, I spent the majority of my time at the middle leadership level and a little with senior leadership. Here’s the more correct ratio:

- Team – 30%
- Middle Management – 60%
- Senior Leadership – 10%

And the most important point here is that I normalized to these ratios as I was coaching across the entire organization and leading it into a state of high-performance. So these were based on the real world dynamics in moving the organization forward.

As I reflect on my most successful coaching gigs, these ratios come through—in coaching, conversations, training, and simply influencing change. The middle management tier in organizations, comprised of team leads, managers, and directors, needs the most help in making the transition. And they’re in the position to do the most with the coaching, helping to sustain and grow overall transformation.

SCENE 3: I AM AN AGILE COACH. I AM AN ORGANISATIONAL DYSFUNCTION

Chris Matts published this wonderfully introspective blog post (<https://theitriskmanager.wordpress.com/2014/03/16/i-am-an-agile-coach-i-am-an-organisational-dysfunction/>) in March 2014. I believe he came to the same conclusion that I did in my ratios — that as coaches, we should spend the majority of our time coaching the leadership teams within organizations. Here’s an excerpt from his post:

This was the point that I realised that I was an organisational dysfunction. Some of the more experienced coaches had suggested I should work for the team doing what was right, rather than work for management. It felt right because management did not have a deep understanding of Agile but I had a stronger feeling that I should be aligned with management who represented the goals of the organisation. The management skills matrix helped me realise that I should not work with the team at all. Instead I should work coaching the leadership of the organisation so that THE LEADERS COULD COACH THE TEAMS. That way, there would be no misalignment. Management would know why they were doing each Agile practice. There would be no disconnect between the teams and management. By training the teams, I am perpetuating a disconnect between the teams and their management . . . I am perpetuating an organisational dysfunction.

I would encourage you to read the entire post.

WHAT'S THE POINT BOB?

As the title implies, I think we (Agile trainers and coaches) are spending too much time with the wrong people.

Instead of taking the easy road (and money) by mostly training & coaching teams, I'd like us to focus on partnering with and training the management tiers within organizations. In fact, I'm starting to think we've been avoiding these folks.

Why?

- Is it because they are in the ugly business of dealing with demanding stakeholders and customers and, as much as we'd like to pretend we understand that world, we don't?
- Is it that our messages, models, and repetitive and simplistic directions don't work as nicely with them? Or is it that we need to show more flexibility and incremental transformation strategies in our guidance?
- Is it that we're afraid of being pulled into their real world vs. our purist views of Agile tactics that apply independent of context?
- Is it because they'll ask tougher questions? And expect us to have relevant, real world experience.
- Is it because it's much tougher to get their time and gain their respect from a change management vs. results perspective?

I'm not sure. But I do know that operating at the team level is "safer" for many of us. It's more secure to "fire up" teams that are sent to us by these very same leaders and managers, and then send them back to their organizations as the primary instigators of Agility.

WRAPPING UP

I know this post may make some in the Agile training and coaching community uncomfortable. It might even anger a few. But I honestly feel we need a "wakeup call".

I think Agile Trainers and CST's should coach more. Perhaps a minimum of 50% of their revenue being generated by coaching and that's across a solid cross-section of their clients.

I also align quite nicely with what Chris Matts was saying in that we coaches need to engage leadership much more in our coaching. Or, as Chris wraps up his post with:

It answers that age old question? Who should go Agile first? The team or the leadership? GIVEN that management want Agile WHEN they hire a coach, THEN the coach should start with management.

So now I have to change the way I work so that I'm no longer a dysfunction. For those of you who know me, you know how hard that will be.

And that includes not allowing folks to bring us in to too heavily engage teams, while not engaging leadership. We need to have the integrity to say no to the easy road inquiries and yes, to the harder contexts that are more leadership focused.

We'll be much better coaches for it AND I truly believe the quality of our Agile transformations will drastically improve.

Which is the point isn't it?

As for me, I want to thank Chris for the wake up call. I will be changing both my training and coaching style and approach within my client engagements. Drastically, probably not. But an immediate and fundamental shift will occur.

Now the question is...what about others?

Stay Agile my friends.

Is It Worth the Energy?

Original Source: <http://rgalen.com/agile-training-news/2016/2/20/is-it-worth-the-energy>

A short time ago I was working with an Agile coach. He was quite experienced and well known in the Agile community. He also held a wide variety of certifications.

We were working together on a project that had, if I were to be honest, quite a few cultural and organizational challenges.

There was one specific individual who always seemed to be the most challenging. My coaching colleague and I were talking about him one day and my colleague was grouching (complaining) about him to me.

After awhile, I asked him if and how he'd approached the situation with the individual. I was looking for a powerful reply where he used the situation to further his coaching relationship with the individual and had a heart-to-heart conversation.

His response though was quite different. He said:

“I didn't mention it at all. Frankly, I didn't have the energy to have a meaningful coaching conversation, so I basically affirmed his behavior, agreed with him, and moved on.”

I responded with an “Oh . . .” and our conversation moved on. He continued to complain about him for a few more minutes and then we went to lunch.

STUCK WITH ME

This conversation has stuck with me ever since.

First of all, I was disappointed in my colleague. I mean the very essence of his job at the time was to have the energy to have just these sorts of conversations.

Not only was it his job, but he also had the certifications, experience, and reputation of someone who could and should have these sorts of conversations. Point being, there were no real excuses for not doing so.

But it did make me think. I began to realize that we all make situational choices every day about how, when, if and for how long we'll engage in “coaching conversations”.

BUT WE HAVE TO MAKE CHOICES...

What are some of the factors that come into play in our interactions?

- **Energy** – I'll start with this one, how much energy will we have to expend to initiate and sustain the conversation?
- **Before** – Have we had the conversation before? How many times? And do we think this moment might be different?
- **Role** – Am I in a role that should be initiating this conversation? Is it my job or have

I been retained to move things AND this conversation is an opportunity to do just that?

- **Timing** – Is it the right locale and timing to have a meaningful conversation? It might be better to wait till later in the day for a private moment to resurface the feedback.
- **Skill** – Certainly comes into play. Do I think I have the skill and experience to have a meaningful and potentially positive coaching conversation? Perhaps someone else who is more skilled should do it?
- **Time** – Do I have the time right now for it? I think this couples with “energy” above.
- **Relationship** – What is our relationship to the individual? Do we know them well, so-so, or not at all? Also, history comes into play here as well.
- **Receptivity** – How well do we think the individual will receive the message or conversation? Sometimes even body language or intangibles (dress) will influence us here.

I’m sure more than one of these came into play in the mind of my colleague before he chose to bypass the conversation. And to some degree, that’s fair.

CRUCIAL CONVERSATIONS

I believe one of my strengths, as a leader and coach, is to usually DECIDE to have these conversations over deferring them.

I’m sort of pit-bullish that way. I don’t shy away from the “hard stuff”. Now sometimes, every once in awhile, I regret this trait. Thinking later that I should have deferred, punted, or in some way ignored the situation and hoped for the best.

Usually these are when the coaching conversations take a lot of time & energy with marginal to unknown outcomes.

And I don’t have the conversation and simply walk away. I’ve always felt that feedback needs to be “verified” after it’s been received to see if folks are actually interpreting your feedback properly AND taking appropriate corrective action.

Here’s a link to a related post on giving feedback that explores this notion a bit more:

<http://rgalen.com/agile-training-news/2015/7/4/with-all-due-respect>

I’ve always felt a bit of professional responsibility for these conversations as well. And I think they’re an incredibly important part of influencing and building a culture.

WRAPPING UP...AND I KNOW WHAT YOU'RE THINKING?

I’ll bet I know what you all are thinking:

Did I round up the energy to have this discussion with my colleague? The answer is . . . embarrassingly no.

I decided that it wouldn’t really do any good and that he didn’t really want to hear it. So my decision was based on time, energy, and receptivity. It was also based on the intersection of my role and his. We were both independent coaches in the organization, so I felt he should have been more self-aware.

All of that being said though, to this day I feel it was a cop-out on my part. Someone should have “called him” on his responsibility to always have the hard conversations OR move onto another role or company. And if I’m confronted with that situation again, I will have that conversation.

When you're a coach, or a leader, it's your job. No matter the reasonable or unreasonable reasons (excuses) you come up with for not having the conversation.

Stay Agile my friends,

Reference

I allude to it in one of the headings, but I believe a wonderful book that aligns with this topic is *Crucial Conversations: Tools for Talking When Stakes Are High*, by Kerry Paterson. I would highly recommend your reading it and it's related follow-on works.

The Agile Coaching Dilemma

Original Source: <http://rgalen.com/agile-training-news/2015/2/18/the-agile-coaching-dilemma>

I've recently been reading about and discovering some Agile coaching firms who have different views towards client coaching. To be honest, I'm struggling to understand and accept some of their perspectives. So, as is often my practice, I thought I'd write something about it to clarify my thoughts and position on the matter.

But first, let me share a story from a close friend of mine in Southern California:

A COACHING STORY

I'm one of the best, most experienced personal trainers on the planet. If you view my website, you'll see testimonials about my:

- Helping transform the health of large groups by running health camps;
- Assisting incredibly famous actors and actresses increase their physical performance to get ready for challenging physical roles;
- Serving as a lead fitness consultant on The Greatest Loser show;
- There's even a rumor that the President will be inviting me to serve on the Council for Physical Fitness.

In a word, I'm one of the country's top Personal Trainers and High-Performance Coaches. My clients approach me because I have a track record of inspiring excellence and significantly improving their health and welfare.

This is what I do and I'm good at it. I get results and quickly.

But lately, I've found that if I push my clients too far or too hard, that they won't engage me for very long. Finding that if I "raise the bar" too far, it damages my relationship with them, but also importantly my revenue stream.

So, I've started to "meet them where they are".

I'll give you a for instance. Bill approached me. He's fairly obese and is just starting to suffer from diabetes and hypertension.

I know I could "push Bill" or better put "inspire Bill" to a fairly high-degree of weight-loss and improved physical health. But it will stress Bill out a bit and I may risk losing him as a client.

So, I'm at a crossroads. Do I coach Bill the way I would normally do it and push him? Or do I change my normal behavior and take an easier road. Knowing that it's not even what Bill approached me for in the first place.

I've decided to only push Bill so far — up to his comfort zone and then no more. I need the clients and the income and if I trade-off a bit of my overall professional integrity, reputation,

and client results – so be it.

PRINCIPLES

In other words, and I'm struggling articulating this, how far do I deviate from my professional principles?

I feel like I'm "faking it a bit" with Bill and I'm not delivering on my capabilities as a trainer. I'm compromising as a trainer and coach. And I'm uncomfortable with it, because it's not really in my DNA.

I got into this business to make a difference. But what are my alternatives? If I "walk away" from clients like this, who lack sufficient commitment, then my revenue will drop by 50%. And I've got a business to run and people depending on me!

So, I'm incredibly torn . . .

Do I hold to my principles and the behaviors that got me where I am today OR do I compromise myself because some folks just can't be held to those same principles?

OK, I MADE IT UP . . .

All right, I need to come clean. This scenario, if you didn't realize it by now, is a fantasy. I made it up.

But it aligns with how I see many Agile coaching and consulting firms behaving with their clients. And it illustrates the "dilemma" that we all face as we're approached by potential clients.

Under the banner of some of the following mantras:

- They're simply too big to apply all of the Agile practices;
- We're meeting our clients — where they are right now and we hope to
- We're not practicing "academic or purist" Agility, our clients live in the real world, and so do we.
- We're moving them along the path to Agility — eventually they'll get there.

These mantras and many others — I see many firms compromising on many of their (the core of) Agile principles: usually not compromising in a heavy-handed way, but more subtly. And I'm convinced that most of their hearts are in the right place and that they are honestly trying to help their clients. And indeed they are, little by little.

But I do think the revenue potential is getting in the way of the decision-making as well. As Agile coaches, we have a responsibility to our clients.

But we also have a responsibility to the Agile Manifesto, the Agile Principles behind it, and to the larger Agile Community to do no harm.

As in my opening story, at some point the fitness coach needs to decide if they're aligned with the principles and performance goals of an outstanding coach or not. In other words, are they pushing their clients as far and as hard as possible for their best interest? In the fitness area, this is accelerated fitness and wellness.

In the Agile arena, this is accelerated adoption that is balanced across team and leadership, while driving significant change and business results. It's not taking a "safe" route, but inspiring the organization to higher insights, practices, and behaviors. And often this involves pushing everyone outside of their comfort zones and traditional ways of doing things.

In both cases, I believe the true measure of the principled coach is their willingness to “walk away” instead of overly compromising their principles.

ANOTHER PERSPECTIVE

Daniel Gullo is a friend and colleague of mine. He’s a Certified Scrum Trainer and a Certified Scrum Coach. He recently went into private practice and is building his own Agile services firm. So revenue generation is important to him and his family right now.

But I find it interesting that he recently wrote the following blog post: *Agile is Not for You* (<http://apple-brook.com/agile-is-not-for-you-2/>). I think it aligns incredibly well with my theme in this article and I encourage you to read it.

WRAPPING UP

All I really want is honesty and transparency.

If your model is to meet your clients where they are and spoon-feed them a few Agile practices — only the ones that they can accommodate or are comfortable with — or ones that are more targeted towards leadership than the teams, then simply be honest and just say that!

Say it in all of your client conversations and in all of your company marketing and branding. Make sure that everyone knows that you’ll be taking a “doing the best we can” approach in your efforts. That compromises and trade-offs, often some that create suboptimal results, will be made.

For example, clearly say that you’re an Enterprise Agile Transformation coaching firm, but you’ll do it slowly, comfortably, and fairly expensively. But that in 3-5 years, perhaps each client will get a return on their Agile investments.

Also be clear that you’ll kowtow to leadership over their teams. Why? Because they are the ones that will be paying the bills and surely you don’t want to push them too far out of their comfort zones. Otherwise, they’ll get upset and look for someone else who will do things “their way”.

I know. I just went a little too far!

As an aside though, can you imagine Mike Krzyzewski of Duke engaging his future recruits in this way?

Come to Duke. We won’t work you too hard. And when you feel like you’ve had enough, of course you can stop and relax. We’re in the business of coaching you in your comfort zone . . . so please, come to Duke.

I wonder what kind of recruits would be attracted to a “let’s work hard to be mediocre” message?

I’m of the feeling that this dilemma will continue in our community. My only hope is that we become more transparent in our coaching principles, both to our clients and to ourselves.

Stay Agile my friends.

What the World Needs is More Prescriptive Agile Coaches

Original Source: <http://rgalen.com/agile-training-news/2014/6/9/what-the-world-needs-is-more-prescriptive-agile-coaches>

I was once working with a peer Agile coach and we were discussing the role of the coach within Agile teams. His view was that it was as a “soft, encouraging, influencing” role. That, at its core, Agility is about the team. And the team in this sense is . . . self-directed.

He also emphasized that taking a more direct or prescriptive approach in our coaching would be anathema to good Agile practices. That it was draconian and dogmatic.

He was actually a leader of this firm’s coaching team, so he had tremendous influence over a team of ten or so Agile coaches. I was one of them and I sometimes struggled with his view and approach.

Now don’t get me wrong. I honestly get the importance of self-directed teams within Agility. I want teams to sort out things on their own. But I also think that we should occasionally provide some direction as coaches instead of always deferring to “it depends”— especially if we’re dealing with brand new teams that don’t have a whole lot of experience. This leads into the whole area of situational coaching, which is where I’m going next.

SHU-HA-RI

A fairly common method for expressing team experience in the Agile community is the metaphor or model of Shu-Ha-Ri. It comes from Aikido and represents three levels of team experience:

1. SHU – Novice, entry level, newbie
2. HA – Journeyman, mid level, experienced practitioner
3. RI – Master, high level, expert practitioner

The metaphor is useful in expressing the situational coaching involved with Agile teams at these various levels. For example, I would expect a coach to be relatively hands-off and simply guiding for a RI-level team.

However, when that same coach encounters a freshly minted, SHU-level team, I would expect them to give the team quite a bit of prescriptive guidance. Also, clearly articulating organizational constraints to the team, for example, helping them establish their Definition of Done.

SELF DISCOVERY

Many of the CST’s have started to present their CSM classes with minimal to no Powerpoint slides. They’re leveraging a style of training entitled *Training from the Back of the Room* (TFTBOTR), which has been developed by Sharon Bowman. The style is mostly focused on

short bursts of discussion followed by hands-on simulation, exercises, or gaming to get the points across.

Of particular interest is the focus by many of our CST's on gaming, where they want team members to learn on their own. Again, while this is useful for some with this learning style, not everyone has this style. And it also assumes everyone being at a certain level of experience.

I guess the point I'm trying to make is that we all have different levels of experience, different learning styles, and different tolerances for this self-discovery approach to learning. At what point does having an expert coach truly directing or prescribing the next 10 steps of your journey help you more than trial and error discovery on your own? And where is the balance?

I'd argue that you need a balance of both, but there is a tendency in the Agile community to lean heavily to the self-discovery and self-direction side of the equation. I want to start challenging that view to a degree.

AND ARE WE BEING "TOO SOFT"?

A famous Project Management consultant and teacher, Neil Whitten, ran a very popular workshop for a number of years. I believe he still runs a variant of it. The title, loosely interpreted was: *The Problem with Most Project Managers—Too Soft!*

His primary premise in the workshop was that project managers lacked the courage to truly engage their teams for what I would call the "hard bits". Things like personal performance, estimate integrity, commitment, providing early feedback on issues, asking for help when appropriate, telling the truth to leadership, taking personal risks, etc.

He pointed out that it was easy to go through the tactics of project management, but that real leadership and maturity was driven from a different place — a willingness and a skill to attack virtually any topic or issue that was standing between the project team and success. That there was a tendency for avoidance of topics that were uncomfortable or difficult to face and discuss.

His main point was that within this space of avoidance lay the success or failure of most projects and that successful project managers had to have the hard discussions and lead from the front.

Now most project managers don't consider themselves too soft. Nor quite frankly, do their teams. But it's where they're being soft that counts. And why am I bringing up this story?

Because I think I want to make the same assessment and then challenge many Agile coaches as being "too soft".

WHAT DOES "TOO SOFT" LOOK LIKE?

I can't speak directly for Neil Whitten, so I'll leave project management alone. However, I can speak for Agile coaching. I do believe we've generally become too soft as a discipline of Agile coaching. There are probably dozens of contributing factors, but I want to share five that come to mind:

- 1. An Unwillingness to "Tell" the Team What to Do** — I see this incredibly often with Agile coaches. The team directly asks them for help and under all circumstances they decline to directly answer the team. Instead, they fall into a pattern saying:

"it depends", asking questions, playing games / simulations, or telling stories as a means of showing the team the way. I often liken this to a "pull request" and frequently I'll directly give an answer to the team. At the very least, I'll give them a few options that I've seen work in similar situations and I'll make a recommendation to them.

2. **An Unwillingness to Step In and Say "Stop It"**—This is an even harder thing to do at times. For example, estimation is something that many teams struggle with. A common pattern is for teams to estimate at too fine a level of granularity. Their hope is that success will surface from the details. But often the reverse is true. That planning at a higher level and sorting through the details as you go is the best strategy. If you encounter a team who is obviously "in the weeds", will you tell them to get out? Even if you've seen this "pattern" a thousand times? I'd say that I would. And I'd like you to consider it as well when you get into similar situations when you know that a team is going to "crash and burn" by using the wrong tactic or practice.
3. **A Lack of Balance in Knowing When to Say When** — Often coaches stay the course in one direction or the other—either they are consistently too hard or too soft. They lack the balance across both of these dimensions. And the teams they coach suffer as a result. I often think that experience comes into play here. Many Agile coaches have little experience in their careers; often less than 5 years of Agile and 10 years of overall software experience. Much of my coaching depth comes from my experience, and that's not simply Agile experience, but my waterfall history helps immensely as well. Don't be afraid to leverage ALL of your experience and don't be afraid to say "I don't know", and ask another coach for help.
4. **A Lack of Situational Awareness vs. Prescriptiveness** — I brought up Shu-Ha-Ri intentionally to illustrate the incredible importance of "situational awareness" when it comes to your Agile coaching — that when you're coaching Shu-level teams, you better be prepared to provide them direct guidance and support. I've found that wrapping the ceremony of reflection or retrospective with situational coaching is a wonderful way to help guide your team. As they are exploring an issue or a challenge and looking for a way to attack it, you can bring up your own stories and advice and get it into play. I also think you can be quite firm here, and yet still let the next steps emerge from the team.
5. **A Fear of Engaging or Getting "In the Game"** — Many formal schools of coaching encourage the coach to stay at a distance. The coach owns the observations, but the coachee, team, or organization owns the action decisions and performance results. There is a fine line between the two. While I honor that view and maintaining some healthy boundaries, I've found that being in the game with the team helps to connect your coaching to the reality of the situation. And it often emboldens the coach to be more prescriptive. I think what I'm saying is that the coach having a stance "as a team or organizational member" is healthy and will draw out more situational prescriptiveness.

WRAPPING UP

I submitted this topic as a session at the 2014 Agile Conference in Orlando and it was selected. I was very, very excited about that and was looking forward to seeing how others in the community reacted to my ideas there. Unfortunately (or fortunately) I was invited to be a part of the Agile China experience the same week and I declined to present at Agile 2014. I'll be doing more research and thinking on this topic in 2014 and will submit it again in 2015.

I'm also making an odd request that I hope some of you take on. I'd like someone to respond to this article with a view to what "too hard" looks like in Agile coaching. I'd love some examples and general guidance and anti-patterns that you've seen in your coaching travels. I guess my point is I'd like to see both sides represented, because I think the truth lies somewhere in between.

Stay Agile my friends, and occasionally try to be more prescriptive in your own Agile team and organizational coaching.

Gene Gendel, CEC



Gene Gendel is Agile Coach, Trainer and Organizational Design Agent. Gene is a proud member of the small community of Scrum Alliance Certified Enterprise Coaches (CEC). Gene's goal is to help organizations and individual teams with improving internal dynamics, organizational structure and overall efficiency. He strives to engage at all organizational levels: senior- and mid-level management, teams and individuals. In his work, Gene uses various methods, tools and techniques to strengthen learning of others and to ensure that teams and individuals gain autonomy after he "coaches himself

out of the job." Throughout his long career, Gene has served small, mid-size and large companies domestically and abroad.

Gene is a well-recognized member of global and local Agile communities, where he influences people via open-space Agile collaboration workshops, coaching retreats, group events and presentations.

Gene strongly supports Scrum Alliance (SA) in its efforts of "transforming the world of work". He is an active member of SA working group of coaches and trainers that have been involved in improving SA certification/education programs by aligning them with natural career paths of Agile professionals: Team Level Coaching Certifications (CTC) (Gene is also one of co-creators of the program.) and Enterprise Level Coaching Certifications (CEC).

Gene's additional credentials are:

- Certified in Agile Leadership (CAL),
- Certified in Large Scale Scrum (CLP),
- LeSS-friendly Scrum Trainer,
- Certified in Scrum @ Scale (S@S),
- CSM, CSPO, CSP, PMP.

Here is the list of Gene's additional focus areas:

- Organizational/System design,
- Enterprise-wide, scaling Agile solutions/frameworks,
- Coaching Leadership, ScrumMasters, Product Owners, Teams

Gene's website:

www.keystepstosuccess.com

Unspoken Agile Topics

Original Source: <https://www.scrumalliance.org/community/articles/2014/july/unspoken-agile-topics>

Introduction

This paper, originally written in February 2013, brings to light some of the least-discussed topics and consequences of “broadband agilization” that currently take place in the industry. The materials of this paper are subdivided into two general sections:

- The first section describes certain impacts that Agile has on individuals and their personal career advancements.
- The second section describes organizational-level Agile impacts that pertain more to client companies that undergo Agile transformation, as well as service-providing vendor companies that deliver Agile-transforming expertise to their respective clients.

The reader will most likely focus on the section that best represents his primary interests and concerns. However, it is recommended that both sections are read in full, as in unison they create a better holistic perspective of the industry changes brought about by Agile-mania.

The reader will be taken out of his comfort zone and forced to think more uninhibitedly and realistically about those aspects of Agile that may not be as obvious and are not explicitly covered in other literature.

Organizational impact of Agile

Peer Pressure

“Change does not necessarily assure progress, but progress implacably requires change. Education is essential to change, for education creates both new wants and the ability to satisfy them.”

– Henry Steele Commager

Problem Statement

According to Wikipedia (http://en.wikipedia.org/wiki/Peer_pressure), *peer pressure* is defined as “influence that a peer group . . . exerts that encourages others to change their attitudes, values, or behaviors to conform the group norms.”

Today companies often decide to introduce Agile practices without thoroughly thinking through why they are doing it, without doing enough due diligence and research to reasonably attest that, indeed, their efforts will bring benefits in the long run. Instead, these companies do so because their executive management has decided that “the time has come,” since there are so many other peers out there that do the same.

For such companies, Agile adoption has almost taken the form of a fashion statement, a way to prove to themselves that they are up to date with others. Such companies care more about keeping up with the mainstream than making a well thought-through and carefully planned step forward of bringing better values, practices, behaviors, and cultural patterns inside their walls.

There is whole array of commonsense Agile transformation readiness prerequisites that these companies either ignorantly overlook or intentionally ignore. These companies go after what may seem to be a very good cause. However, it is nothing more but a chase of the “status quo.”

In majority of cases, such poorly substantiated motives for Agile adoption bring about failure, and since there is only one chance to make a first impression, any future attempts to reintroduce Agile at a later point, even when conditions might become more favorable, usually meet resistance and very little support. Trust is lost as nobody wants to repeat the same mistakes twice.

Discussion

“Going Agile” should never be a final goal. Agile is just a way to get to something much more measurable and tangible. For example, achieving near-term and long-term economic benefits, ensuring cost-effectiveness and higher return on investments (ROI), adjusting corporate cultures and working environments in ways that make companies a more desirable place to work.

Many firms proudly announce that they have been undergoing Agile transformation without accepting the fact that in order to truly appreciate the benefits of Agile tools and techniques as the mechanism for more effective product development, firms also must adopt Agility in their structure and culture. The former is just not possible without the latter.

Specifically, if an organization does not have in its plans to fundamentally adjust its conventional hierarchical structure, flatten its convoluted reporting lines, remove redundant roles that do not really contribute much value to the overall process, and trim wasteful activities, then there will be no cultural shift.

While using the Toyota Production System (TPS) as an example in their studies of Lean, Tom and Mary Poppendieck have clearly identified seven types of waste in Agile Product Development.³ The third item on their list of wasteful activities is “Extra Processing.” If we think about any process in terms of how many people are involved in it, then it begs the question: Should removal of wasteful processing also remove wasteful processor(s)?

Just as an assembly line automation makes a lot of manual labor unnecessary, eliminating unnecessary steps in a process makes the performers responsible for those steps also unnecessary.

This supports the claim that in order to successfully transition to Agile, companies should be willing and ready to reorganize their cohorts in ways that require trimming off what is

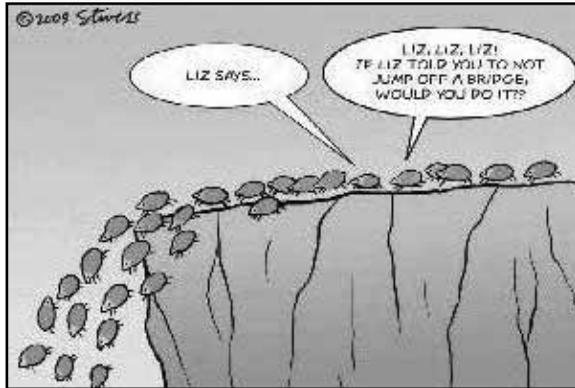


Image courtesy of <http://www.markstivers.com>

no longer needed.

In his writings, Craig Larman⁵ alludes to organizational waste management by referencing not just wasteful and counterproductive processes and individual behavioral patterns but also certain organizational layers and individuals that cause retardation of the Agile process. While being a big proponent of flattening organizations in general, he explicitly refers to certain mid-level management that should be removed due to their lack of value.

There are some other pitfalls that are more frequently seen with larger, enterprise-level organizations:

- Inability to perform cost/benefit analysis to determine whether the adoption of an Agile framework is actually monetarily beneficial for a company. Since larger companies typically do not release to production as frequently as smaller ones, it is not always easy to map end-client satisfaction and demand, and subsequent revenue increase to changes of product development approaches. This delayed cause-and-effect is due to longer time to market and prevents executive management from seeing results of Agile transformation soon enough to decide whether it is worth continuing the experiment. Executives do not use a “stop the ship” approach to objectively analyze what has been accomplished in order to decide whether it makes sense to continue. If things do go south, by the time executives realize this, the damage is too high to be dealt with silently, behind the scenes. When the political current gets too strong, as it does in the majority of cases, fighting it and acknowledging that the latest and greatest add-on to a company's strategy is not working is not something that the top echelon of executives can do easily.
- Inability to take into account existing business relationships with third parties whose operational models, processes, and functions adversely impact Agile adoption. Specifically, dependencies on third-party product development vendors who do not practice Agile, do not deliver incrementally, and -- what is even more alarming -- fundamentally are not equipped for transparent two-way communication (they prefer everything in writing, sealed and signed) is very costly. This can negatively impact a company's ability to produce its own deliverables. Such relationships and dependencies must be thoroughly reevaluated and, if necessary, terminated.
- Inability to develop standardized techniques or mechanisms to measure levels of Agile maturity across multiple organizational verticals. Although developing universal best practices for multiple teams, even within the same organization, is not expected (it would also contradict the idea of decentralized control and frequent inspection and adaptation -- something that is required by Scrum), the ability to tie low-level (local, single-team) Agile metrics to global performance indicators is possible and even desirable, as ultimately every company measures its profit and loss by using universal units (currency).

Instead of attempting to do “big bang,” top-down Agile transformations of multiple teams, projects, and programs at the same time, it is much more advisable for companies to start small, to consider a pilot Agile project to gain small, quick wins, and then gradually proceed with Agile adoption by sharing knowledge and lessons learned laterally: from team to team, from project to project. It is important, however, that executive support and buy-in come all the way from the top-executive level -- what is inevitably required for a lasting cultural shift. Executive-level coaching is required for this to be a success.

Numbers Do Lie

“Nothing in education is so astonishing as the amount of ignorance it accumulates in the form of inert facts.”

– Henry Brooks Adams

Problem Statement

In most of reference literature, the word “metric” is defined as the system of standard or measurement. In the conventional world, the notion of metrics analysis is frequently associated with establishing success/failure criteria, progress indicators, and benchmarks.

Although the ability to properly analyze and communicate Agile metrics data still serves its meaningful purpose for Agile teams, as it gauges them in their journey toward maturity, for large-scale Agile transformations at enterprise level (not so much small- and mid-size organizations), the improper use of Agile numbers is common and dangerous. The ability to incorporate Agile data into a broader picture and integrate it with other enterprise-level measurement tools, techniques, and analytical facts is frequently lacking.

Discussion

If Agile transformation is driven from the top of an organization but appropriate training is not provided to executives in a timely manner, then their expectations are not properly set and this leads to misjudgment and poor decisions.



Image courtesy of <http://www.processexcellencenetwork.com>

The problems vary and include constant pressure on workers and deterioration of morale, mistakenly (in a rush) selected corrective actions, and/or making things “look pretty” by falsely communicating unachieved progress further up the chain of command and to the rest of organization in order to preserve personal credibility and reputation.

One of the most common misinterpretations is of metrics obtained from Agile collaboration tools. The words of wisdom in IT are: “A fool with a tool is still a fool.” This wisdom has proven itself many times. One of the most frequently misinterpreted and misused metrical units is velocity — specifically, what it measures and what it depends on.

Here are some most common misjudgments regarding velocity:

- Velocity reflects how much time it takes for a team to complete work. No consideration is given to the fact that in order for complexity estimation to be accurately translated to time, the same team must point stories and stories must come from the same PO, who has the same writing style and is able to write “sizably.” This is wrong.
- Without prior normalization of story point estimation across multiple teams (deriving a conversion factor to normalize 1 point of one team against 1 point of another team), it is OK to compare velocity of one team to that of another team. This is wrong.
- Over time, velocity should continuously increase and if it does not, it indicates that a team has stopped improving. This is wrong.
- A team’s velocity can be increased by increasing the number of team members; this is an effective way to increase velocity. This is wrong.
- Resource rotation (on/off a team) to ensure cross-training and knowledge transfer

outweighs the importance of keeping a team together and does not have impact on velocity. This is wrong.

- Reestimating work mid-sprint is acceptable, if it can provide a correction to inaccurate estimations at the beginning of a sprint. This is wrong.
- Attributing a certain percentage of a team's committed velocity to an individual team member, based on the contribution of that individual to any particular story, is an objective way to measure individual productivity/performance and a reliable way to gauge overall team velocity. This is wrong.

This is how some of these false interpretations come about:

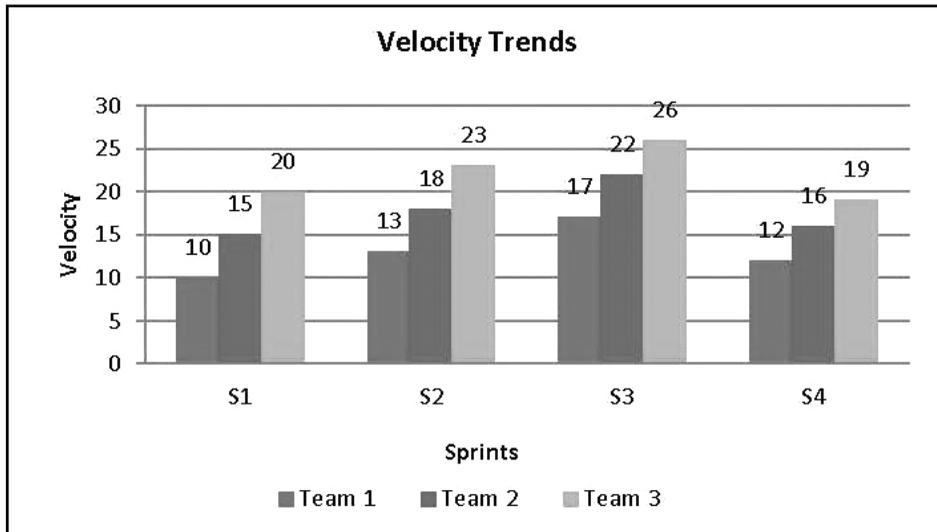


Figure 1: Comparing un-normalized velocity

Figure 1 above illustrates an example of three separate teams being compared against each other in terms of their velocity. Such a comparison does not hold value for the following reasons:

- Team size and, subsequently, its capacity, most likely varies. This means that the man-hours of one team are not comparable to the man-hours of another team.
- The estimation scale of each team is different. In order to compare a story point of one team with that of another, a conversion factor must be derived to understand what each team's story point really means in terms of ideal hours.
- Finally, Agile maturity of teams might be different. Although a team's maturity can be related to a team's productivity/output, comparing teams that are novice to Agile with those that have been in operation for a while and have developed some cadence is not objective.

Here are some graphic illustrations of how such improper judgments originate:

Figure 2 below illustrates an example of a false expectation that, over time, the velocity of each team will indefinitely increase. Team 2 (brown) appears to have the velocity trend that aims at infinity. Team 1 (blue), on the other hand, has reached a plateau and is no longer increasing. It is false to assume that the velocity trend of Team 2 is better than the velocity of Team 1. Yet, frequently, this is exactly the conclusion management derives when they are presented with multi-team velocity trend charts.

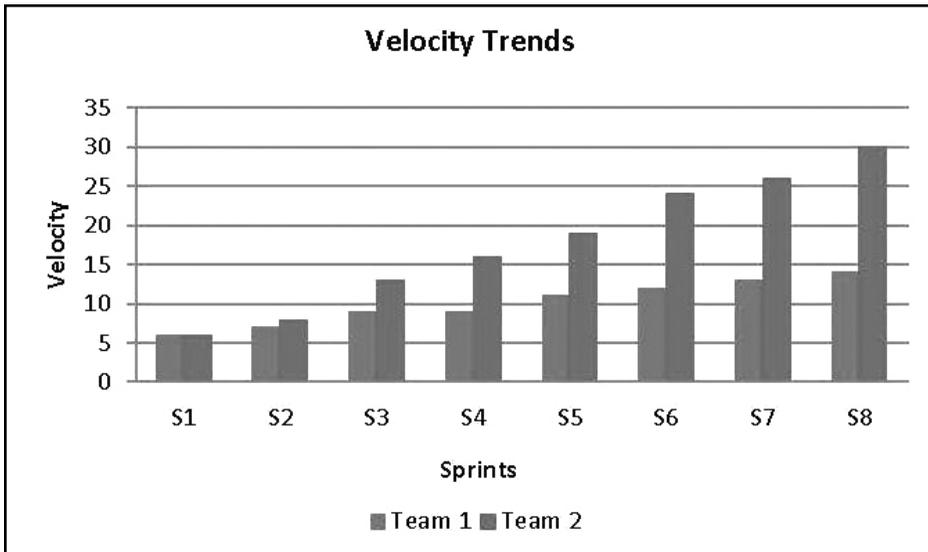


Figure 2: Expecting continued velocity increase

Such ever-growing to supersonic velocity of Team 2 is not realistic. Increasing velocity to a certain reasonable level and then sustaining it would be much more reasonable. Having a steady velocity would also make forecasting and forward planning more reliable.

Figures 3a and 3b below illustrate how a team's sprint velocity can be mistakenly attributed to individual contribution.

Note: If added diagonally and horizontally, percentages add up to "1" (100%)

	Stories			Total
	Story A SP	Story B SP	Story C SP	
	3	5	8	16
John (% of time on Tasks)	0.45	0.25	0.3	
Jeff (% of time on Tasks)	0.25	0.5	0.25	
Jim (% of time on Tasks)	0.3	0.25	0.45	

Figure 3a: Measuring "individual velocity"

Figure 3a shows how three individual team members spend varying amounts of time (in percent) on each individual story (later, accepted). Such work distribution could be very reasonable as each individual may need to contribute differently to each story, based on his type of functional expertise and type of work required for each story.

Figure 3b, on the other hand, shows a very unreasonable calculation and, subsequently, a conclusion that is frequently made by conventionally thinking management, especially if an environment strongly supports the idea of individual performance. According to this figure, each team member is assigned a certain amount (even fractional) of story points, based on his percentage of effort contribution to each user story. The numbers are derived by multiplying individual percentage contribution and number of story points each accepted user

	Points Per Story by Person			Total "Individual Velocity"
	Story A	Story B	Story C	
	Points =3	Points =5	Points =8	
John (% of time on Tasks)	1.35	1.25	2.4	5
Jeff (% of time on Tasks)	0.75	2.5	2	5.25
Jim (% of time on Tasks)	0.9	1.25	3.6	5.75
"Individual Velocity" per Story	3	5	8	16

Figure 3b (Measuring "individual velocity")

story is worth. For example, John, who performed 0.45 (45%) of work (in hours) against Story A (3 story points), established his own “velocity” of 1.35 story points against Story A.

Such an approach is wrong, and here is why:

To achieve higher productivity and cohesiveness, Scrum team members are expected to swarm -- work collectively on the same story or sometimes even a single technical task to achieve common success. Linking any single piece of a team’s work, successful or unsuccessful, to individual contribution and performance is inappropriate and subjective: it prevents teaming and collaboration as well as makes team members worry about individual achievements more than about overall team success.

Another reason why the above described calculation is a fallacy is that although each team member cumulatively contributes the same amount of time to sprint work, based on an individual’s functional skill set, each person may spend time (capacity) differently against differently pointed stories, which means that the multiplication factor in deriving individual velocity is not constant from the beginning.

Another misuse of the velocity concept is splitting up partially done stories at the end of a sprint to produce “partial” or “conditional” acceptance by the PO. By doing so, higher team velocity gets fabricated.

While breaking the concept that each story must be an independent and deployable piece of functionality with intrinsic business value, such numbers-cooking and story-point chasing will prevent a team from establishing a reliable velocity and doing accurate sprint forecasting and strategic planning.

There are also some other inaccurate interpretations of Agile metrics that, for the most part, have to do with leaders’ lack of understanding of the economic principles behind product development. Among others mistakes, the two that are very commonly observed are around Capacity Utilization and Work in Progress (WIP).

For example, forcing teams to maximize their Capacity Utilization by increasing individual and team workload to close to 100 percent, as it is frequently seen in “high-performance” organizations, especially when using offshore resources, is a fallacy.

Such an approach forces teams into working without any slack time and, therefore, deprives teams of any chance to improve processes. By the same token, pushing teams into making aggressive commitments during planning and starting a sprint (e.g., in Scrum) with an initially unrealistic amount of work causes end-of-sprint failures. All of this results in a team’s diluted focus, excessive multitasking, making irrational decisions, and ultimately producing code of very low quality.

Expecting high Work in Progress (WIP) by having executives constantly question teams about the amount of work in flight is yet another fallacy.

By applying such orthodox beliefs that everyone should be preoccupied with their own work at all times, and promoting the idea that a high amount of work in progress is a sign of high effectiveness and productivity is bogus. This contradicts principles of one piece of work flow, queue size management, and capacity-utilization principles. It also conflicts with the concepts of collaboration and swarming that are strongly supported by cross-functional feature teams.

Some convincing studies about capacity utilization and its effects on productivity were done by Donald Reinertsen.^{7,8} They are worth mentioning here because they can be tightly coupled with one of the Agile product development tools: Kanban.

Based on Reinertsen's Principle of Part-Time Resources (using part-time resources for high-variability tasks), maximizing the load of key resources with high-priority tasks is dangerous if more high-priority work is expected. Individuals who are highly loaded with high-priority work have low surge capacity, which is extra bandwidth that they can use against newly arrived high-priority work. What this means for Kanban teams that work on production support issues with different levels of severity is that the lack of surge capacity prevents a team from being able to switch to high-severity issues when they arise.

For example, imagine a Kanban team that has work of L1, L2, and L3 levels of severity moving through the same queue:

If a worker is always fully preoccupied with L3 work, his surge capacity very limited (literally, to his lunch hour), and this prevents him from picking up any additional L3 work, should such work enter a queue. This is particularly dangerous, especially if other workers that are fully preoccupied with L1 work (though having much higher surge capacity) have insufficient skill sets to handle incoming L3 work. It makes much more sense to optimize individual workload in ways that everyone, especially highly skilled specialists, have enough surge capacity (slack time away from L3 work) to be able to react to suddenly incoming high-priority work.

Challenges with Agile leadership

“Divorced from ethics, leadership is reduced to management and politics to mere technique.”

-- James MacGregor Burns

Problem Statement

Today, the most widely recognized Agile leadership role (above an individual team level) is an Agile coach. Typically, Agile coaching is delivered by an external consultant who either engages with a company (client) directly and independently or represents a specialized Agile coaching/training firm that deploys him on site.

Recently, companies began introducing internal Agile coaching practices by way of attracting external consulting talents and then converting them into full-time employees, and/or by retraining existing company employees to transform them into Agile coaches (native coaches).

In the former case, the following two questions usually come up:

- Does engaging with a reputable Agile coaching consulting firm guarantee a top-notch Agile coach-consultant who will be deployed on site?
- Once engaged, what are the odds that a consulting firm, intentionally or unintentionally, positions itself in such a way that its financial benefits from the engagement become more of its focus than value delivered to a client?

In the latter case, when internal coaches are used to help an organization with Agile transformation, the concerns are somewhere different:

- Internal coaches that have not been exposed to the outside world (other industries, other corporate cultures) tend to have views that are narrowed to only what

they have seen at their own companies (structure, culture) and, therefore, their ability to comparatively analyze organizational problems is significantly hindered.

- Internal coaches, native or “naturalized” (defined further below), being full-time employees of an organization, are subject to evaluation, scrutiny, and performance measurements that are in conflict with what Agile culture needs. Such strict limitations, obviously, make it difficult even for the best coaches to provide (uncensored) reflection of a company.

Discussion

When a company decides to procure an external coach and convert him into an employee (get him naturalized), a coach becomes subjected to the same type of evaluation and scrutiny as a native coach, or for that matter, as any other employee.

What does it mean for a role that historically is meant to be held by an independent, neutral third party?

An internal coach, unlike an external coach, cannot as freely reflect upon a company’s ability to help itself by acknowledging its own problems and finding its own ways to resolve them. Now, a coach is part of a company and the expectations of him are different. What an internal coach says about his own employer and the conclusions and recommendations a coach gives to his own employer will inevitably influence how an employer treats the coach. A coach’s job is discovering/exposing organizational pain points, asking powerful, and at times uncomfortable, questions, as well as giving bold reorganizational recommendations — a coach’s ability to do so becomes a hostage to his or her fear of becoming the subject of criticism and reprisal. As coaches become (or remain) part of an organization, they are naturally forced to move away from being servant leaders to becoming more of personal achievers and politically correct commanders and controllers.



“Bound” by Connor Tarter (Attribution-ShareAlike 2.0 Generic (CC BY-SA 2.0))

Back to the *external* coaching consulting model. One of the clearest representations of the Agile coaching dilemma has been described by Dan Mezick in his book *The Culture Game*. The author vividly paints how important it is to define entry and exit criteria for every coaching engagement, as well as its duration, before it begins in order to avoid excessive monetary transactions and long-lasting codependency between a client company and an external coach — an indication of unethical coaching.

Today, unfortunately, most companies still rely on external Agile coaching expertise blindly, without questioning its effectiveness. Large-scale coaching engagements are frequently secured based on personal relationships, where SOWs get signed and monetary transactions take place, not where transformation takes place and value gets added.

With external Agile coaching, when Agile transformations get done in one “big bang” and there is suddenly a high surge in Agile coaching demand by a client company, the Agile transformation company runs a fire drill and tries to immediately produce additional coaching staff, by turning to the marketplace and procuring independent coaches from the street and then reselling them to the client as “their own.” This certainly does not guarantee to a client the good quality of a coaching resource, and it most certainly does not render such resource at a true net cost (the mark-up for these types of engagements is usually pretty high).

Another scenario is when a coaching role gets occupied by an individual with great coaching skills but very little practical, hands-on experience within Agile. Typically, such situations arise when a person comes from a completely different area of coaching (personal coaching, spiritual coaching, career coaching, life coaching, etc.) and effectively uses his “soft”/people facilitation skills to compensate for very superficial subject matter expertise.

Since Agile is primarily about product development, ideally the person who steps into the Agile coaching role should have some technical or at least semi-technical background. This would help him or her better understand and appreciate product development issues, and therefore provide better advice and earn a higher reputation.

Going by the same logic, when a company decides to build its own Agile practice, it is very important that selected individuals do a good share of observing and shadowing more experienced Agile coaches before taking their own initiative. Co-coaching with more experienced peers helps a novice coach not only capitalize on his understanding and practical know-how of Agile mechanics but also adopt proper behavioral patterns of being a servant leader and enabler, not a commander and controller -- something that is often seen when a coaching role gets filled by an internal person who was previously an authoritative figure.

Agile Impact on Individuals

Struggle for Personal Adaptation

“Today a thousand doors of enterprise are open to you, inviting you to useful work. To live at this time is an inestimable privilege, and a sacred obligation devolves upon you to make right use of your opportunities. Today is the day in which to attempt and achieve something worthwhile.”

-- Grenville Kleiser

Problem Statement

Agile affects professional careers and personal lives. So let’s pause here for a moment and make an important distinction: Agile was initially introduced as a way to develop products. Its purpose was not to manage individual projects, as is incorrectly perceived by those who are familiar only with traditional software development. Nor was it a “plug-in” into any other method or framework that companies use. The purpose of Agile was (and remains) to get a product of the best quality to a client in the shortest time frame, at the lowest cost possible.

For many individuals, Agile is a great way to explore themselves, to reveal and further develop their individual potential. Agile favors innovative thinking, helps merge the gap between creative art and the science of technology, and assists in gaining the freedom of making choices and developing the Kaizen culture.

Since Agile originated primarily as the mechanism for product development, individuals with skills that are required for product development are able to adapt to Agile relatively quickly.

On the contrary, for those individuals whose skills are not directly related to product development processes, Agile adoption presents a significant challenge. Such individuals cannot effectively contribute to day-to-day activities needed for Lean product development, they don’t easily fit into the flatter organizational structure required by Agile, and they cannot

adapt to the absence of the command-and-control environment that prevailed under previous working conditions.



Image courtesy of <https://www.actonmba.org>

environment, a developer is given many more opportunities than in a non-Agile environment to explore his intellectual capacity, while thinking outside the box and coming up with innovative solutions. No longer does a developer obediently execute against frozen business requirements that were most likely put in silo by a business analyst, with minimal or no input from technology and with minimal or no exposure to real end users. In the latter case, by the time a developer starts coding, requirements are most likely stale and out of date. This ultimately leads to change requests and, most likely, to product changes when they are the most expensive to make.

In Agile, a developer has direct communication with a “buyer” (end-client or an empowered proxy, the product owner). A developer now has an opportunity to look at each business requirement (usually a user story) individually and offer a unique, at times innovative, technical solution with a very short feedback loop (response) from a client. In case of a positive feedback loop, a developer is encouraged and happy to claim a small credit for his win and a job well done. In case of a negative feedback loop, a developer has little damage control to do, as the amount of rework required is usually minimal.

What developers do find challenging, however, as they transition from a more conventional environment to Agile, is that they are not always able to work “on par” with other developers. In Agile (let’s take the Scrum framework for example), developers on the same team might be at different levels of seniority and, even more undesirable, unevenly positioned with respect to each other in the same organizational hierarchical structure. This is a problem, as the lack of equality among team members prevents them from having effective collaboration.

But all in all, for a skilled developer who is willing to cross-train further and become a true T-shaped person (specialist in one field plus a generalist in one or more other fields), Agile is a land of great opportunities for personal and professional growth.

QA

The situation is even more straightforward with QA. Let’s make a note here about one very important precondition for scalable Agile: automated testing coverage.

If manual testing still predominates over automation, development will stall and plateau as

Discussion

Agile is meant for true doers. If we recall Donald Reintersen’s^{7,8} discussions about product development, the closer an individual is located to the automated production line, the more value he brings to the process. In terms of software development, this might be translated as follows: The closer an individual is to a code base, the more value he brings to product development — and vice versa.

Developers

Let’s take a look at a developer. In an Agile

soon as manual QA is not able to keep up with development. Ideally, test automation should begin with the first line of code or, even better, it should come before coding begins (e.g., TDD/ATDD). Success in Agile is not possible without test automation.

In Agile, QA involvement begins much sooner than in Waterfall, where QA gets to see a product only when development is practically done and when the discovery and repair of bugs is the most expensive. In Agile, QA's role is elevated significantly, becoming a much more reputable role. The discouraging notion that we hear at times, "A QA person is someone who is not good enough to become a true developer," is no longer valid in Agile, as QA is now rightfully considered a member of the development team and not just someone who manually executes only by following a handwritten test case.

There is one big assumption, however: that a QA person is willing and able to become and/or remain technical. Any automation test tool requires some coding skills, and since true Agile cannot exist without test automation, the QA person should be comfortable with coding. There may be no need to become a full match to a senior programmer, but QA does need to come closer to a developer in terms of technical savviness.

This might present a challenge to those QA people who have done manual testing only. In Agile, manual testing is only temporarily valuable during initial sprints, when code base is limited and there are not too many features to test. Over time, as more code gets produced and more functionalities become available, manual end-to-end testing cannot keep up with development. Therefore, we need a machine.

Similar to developers, the ability to work "on par" with other team members may present a challenge. Here, the adjustment for QA people is more psychological and cultural than functional -- they have to be at the same level with other team members, regardless of their current position in the organizational tree.

Overall, if we assume that cultural adjustment does not present a serious challenge for a technically predisposed QA person, Agile also presents an array of opportunities in terms of gaining more hands-on knowledge, professional respect, and team recognition.

BA

The role of the business analyst is clearly articulated on the www.iiba.org website. According to the International Institute of Business Analysis, the role of the BA includes various types of analysis: systems, requirements, data, process, business intelligence.

Since the discussion of this paper focuses on Agile, and given the fact that the main intention of Agile is to improve product development practices, our main focus here is on BAs who participate in product development and serve the purpose of a primary conduit between business and technology.

Since in Agile product development the intent is to bring closer the business community (end-users) and technology (feature teams), if BAs want to stay close to the product development process and survive organizational flattening, they have to position themselves in one of the following ways:

- Assume the role of product owner
- Assume the role of product owner-proxy (if such a layer is justified)
- Embed with a feature team as team BA

In order for a BA to be able to assume a product ownership role, his level of authority and

executive decision-making power must be significantly increased. Today, in conventional settings, even very senior BAs cannot make final business decisions on their own, as they have to seek approval of more senior staff, including business stakeholders, sponsors, etc. Even in instances when BAs are able to formulate decisions based on the inputs of many, the cycle time from the moment the BA raises a question to the moment he is able to derive a conclusive decision and communicate it to IT is way too long to support an Agile development pace that is based on a short cycle times.

Looking at the situation realistically, it is highly unlikely that the BA will get empowered to a level that he could be rightfully considered the PO. For this to happen, most likely, the BA would have to jump a few hierarchical levels, something that does not happen frequently. Such BA empowerment is even less expected at large, enterprise-size companies than at small/mid-size companies, as reorganizational decisions take place much more quickly and a flatter structure is more natural in the latter. Therefore, although having a BA become a PO is possible, at most enterprise-level companies it is unlikely.

What is much more likely to happen is to have a BA assume the role of PO-proxy — a role that is sometimes introduced to help the PO with his responsibilities. Introducing the PO-proxy role is much more common at large, enterprise-size companies than at smaller ones where a company's primary line of business is software development.

You may ask why. The answer is simple: A company that primarily generates its revenue from building and selling software to external clients cannot afford to have a multilayered product ownership structure. The risk of miscommunication and delayed response is just way too high. The PO role is taken much more seriously at smaller/mid-size software development companies. It is a full-time job and whoever takes it gives it his full focus.

At larger companies, however, having a BA or BA-like person assuming a PO-proxy role is much more common; larger companies are more tolerant of having PO-proxy roles. Meanwhile, the role of actual PO (head PO or chief PO) is given to a person with more stripes on his shoulders.

Here is a typical scenario:

The PO role is given to someone who is positioned high in the organizational food chain — someone who is entitled to make final business decisions. But in the majority of cases, such a PO is more of a political figure who neither fully understands nor is being held accountable for performing his duties as PO. He is not so much a decision maker as a decision “signer,” whereas decisions are recommended by someone else. This “someone” is typically a PO-proxy, a lower-ranking role that is almost immediately introduced after selecting the PO. The PO-proxy role becomes a buffer layer between the PO and real work.

Although there are instances in which the model of a single PO plus multiple PO-proxies makes sense (e.g., the PO is responsible for a product that is being built by multiple feature teams, where each PO-proxy supports each individual team), in general such additional organizational layers create unwanted risks: miscommunication, misunderstanding, and increased cycle time.

In his book *Agile Product Development with Scrum*, Roman Pichler clearly describes how having BAs stepping into PO-proxy roles creates multiple problems that ultimately lead to “decrease of productivity and morale.”²

Therefore, although becoming a PO-proxy is much more realistic than becoming the PO, it still does not provide a very effective solution for accommodating BAs, as the volume of BAs that each organization harbors today by far exceeds the number of available PO-proxy openings. This is a very basic supply-and-demand dilemma.

PO "Candidate"

Let's face it, for a business analyst (or for someone at the same organizational level) to step into a PO's role is nothing but a great opportunity for career growth. It is a step up into the spotlight, gaining more visibility and authority, being presented with more opportunities to network and form useful professional relationships. Today, there are many BAs who are asked to do the heavy lifting for POs, including writing stories, backlog grooming, communicating with feature teams (especially offshore teams) — pretty much everything except making final decisions. This creates a situation in which BAs do a lot of heavy lifting for little recognition.

Elevating BAs to PO-proxies (still much more realistic than becoming the PO), deputizing them to be not just backstage servants but rather front-stage leaders, creates a rewarding situation for them, whereby assuming a more important organizational position becomes very attractive.

Let's take a look at the opposite situation: An individual who already has a high-ranking job with a company is asked to step into the PO's shoes.

Such an individual already has a plenty of visibility, authority, and, most certainly, lots of day-to-day responsibilities. This individual's job has already been defined in "pre-Agile" terms with clearly formulated success/failure indicators, and, crucially, a compensation structure. Naturally, such an individual will not genuinely embrace the additional PO role unless the following conditions are met:

1. His other day-to-day responsibilities are minimized.
2. His compensation is increased to justify for additional efforts required to perform the second job.
3. There is a hybrid of the first two conditions: reasonable workload, reasonable compensation, no significant loss of organizational positioning.

In a majority of cases, companies attempt to fulfill the first condition but complete it only partially by simply doing an internal reorganization and appointing an individual for the PO role without removing his existing responsibilities. There will always be some resistance along the way. This is why:

For a product manager or a key SME/business user to have more direct interaction with technology usually means performing "dirty" work.

For a product manager or a key SME/business user to step into a PO's role sometimes means stepping down in the organizational tree and giving up direct reports — something that might be required to avoid conflicts of interest.

Regardless of overall workload for a newly baked PO, the type of work required by the role and how this work ties to organizational positioning and monetary rewards (e.g., potential bonuses) is usually the reason for low support.

Under the second condition (the likelihood of which, by the way, is not high at large organizations since salaries and bonuses are tied to organizational levels), yet another likelihood for

failure is that no extra money would compensate for the extra time and energy that a worker needs to spend on doing a second full-time job. It is highly unlikely that an individual would be able to sustain twice the workload without having it affect his personal lifestyle. It is also unlikely that a company would be willing to increase an individual's compensation twofold to pay for a double effort. And as has been proven many times, “highly compensated heroics” never have long-lasting effects.

It seems that the only potentially workable option would be to create hybrid conditions under which, on one hand, the PO feels safe that his role within an organization did not depreciate (although the workload would decrease) and, on the other hand, he was able to give the required time and attention to Agile teams as needed. At the same time, the PO's compensation increase and rewards would have to be within reasonable terms to compensate for significant additional work, yet not cause serious exception to a compensation formula used by a company.

In his book *The Art of Product Management*, Richard Mironov¹ describes situations when the role of product owner in Scrum is fulfilled by a person whose day-to-day duties and responsibilities resemble that of a PO: product manager. The product manager is a conventional, outward-facing role of a product business owner. Of all potential candidates for the PO role in Scrum, the product manager seems to be the closest to the PO role.

Mironov goes into detail, outlining how the two roles (PO and product manager) differ, specifically stressing the fact that the speed of crashing/failure is much higher for a PO than it is for a product manager. This is due to the fact that things move much faster in Scrum and time frames to observe antipatterns and shortcomings are much narrower.

The most important thing that the two roles have in common, and the main reason for the lack of success: lack of engagement.

Project Manager

One of the most questionable roles in Agile product development is that of the project manager.

Is this role still required? This topic is controversial and causes a lot of discomfort when raised. For many PMs, the uncertainty about how their jobs will be impacted by Agile is the reason why Agile meets resistance in the conventional PM world.

Agile adoption presents different challenges for technical managers and nontechnical managers. It is important to make a note of this distinction.

For mid-level technical managers, the dilemma is primarily psychological and behavioral. Technical managers are expected to have hands-on expertise and, if needed, should be able to roll up their sleeves and produce technical work relatively quickly. This is exactly what counts in Agile product development — producing tangible technical work. The main challenge that technical managers face is their ability to let go of authoritative power and control of their subordinates. In Agile, tactical technical decisions are to be made at a team level, not at a technical management level. Psychologically, loss of such centralized control creates a problem — it is discomforting. The situation may worsen if a technical manager needs to become a member of a feature team, where he starts working side by side with individuals who previously reported to him.

As mentioned in the discussion about POs, organizational flattening and the adoption of an

Agile framework may translate into a loss of jurisdictional power for some, and technical managers are not an exception.

Nevertheless, this does not create a “potential job loss” situation for technical managers, as they are always able (or at least expected) to fall back on their primary technical skills and to integrate with feature teams. Alternatively, some individuals can get promoted from mid-level technical management to more strategic technical leadership, where they are not be so much involved in tactical work at the team level but are responsible for more strategic decisions and resource planning across multiple teams.

But again, just like the space for “upraising” BAs to PO-proxies is limited, so is the space for uprating mid-level technical managers into more senior positions: Supply and demand rules still apply.

It is important to note that the notion of senior technical leadership does not go away when Agile is introduced, as senior technical management is still needed. It is the abundance of mid-level technical management and the redundancy of their work.

Things are much more different for non-technical managers.

In Agile, let’s take the Scrum framework as an example. The responsibilities of the project manager are evenly distributed between the PO and the team. The PO is now responsible for all strategic planning: product vision, product road map, time lines, budget, and scope (remains flexible most of the time). A team is responsible for all tactical activities: sprint planning, story estimation, task breakdown, work assignment and work flow management, various team ceremonies, etc. A ScrumMaster, selected by a team (the ideal case), usually picks up various team logistics, resolving inter- and intra-team impediments, brokering and facilitating ceremonies, negotiating with the PO, protecting a team from undesirable external influence, and escalating problems to executive management.

So is there anything left to do for a mid-level non-technical project manager in an organization that gets leaner and flatter as it undergoes Agile transformation, lightens its processes on all fronts, and gets rid of all its “procedural” waste? It all depends on how easily a non-technical PM is able to adjust.

Among other less apparent elements that are required for mid-level non-technical PMs to stay afloat in an Agile-transforming organization, the two main adjustment requirements are:

1. Mental shift away from command-and-control behavior
2. Ability and willingness to pick up additional technical skills that would make the PM more valuable in a product development process

The first is all about behavioral patterns. It is about accepting that a group of skilled professionals, when empowered, can make decisions about their own work better than any outsider who is not doing the actual work, especially if such an outsider is not qualified technically. Since Scrum (we continuously use this Agile framework as an example, as it is the most structured one of all Agile frameworks) implies self-direction and self-governance, any attempts to force anything upon a feature team will have adverse effects on both parties: on one hand, it will deteriorate Scrum and stall evolvement of a Kaizen culture, and on the other hand, it will marginalize a person who attempts to act as enforcer even further from where the real action takes place.

At this point, it is worth mentioning one very common anti-pattern that is often observed in large organizations as they undergo Agile transformation: The ScrumMaster role automatically gets filled by a former PM. Is this default assignment proper?

Automatic sanctioning of PMs with SM responsibilities may cause more harm than good, and this has proven to be the case on multiple occasions. Having PMs go out and get certified as ScrumMasters is not sufficient. If a PM mind-set remains, the person will never become a ScrumMaster. Unfortunately, mind shifting is not something that any certification can change. When the PM steps into the SM role but continues using his command-and-control tactics, it demoralizes the team and becomes its biggest impediment. The situation becomes even more dangerous if a newly baked SM who used to be a direct manager of other (one or more) team members now becomes their SM. Even if organizational reporting lines are removed, psychological dependency frequently remains and prevents team members from thinking and acting freely.

In his book *Scaling Lean and Agile Development* (coauthored with Bas Vodde), Craig Larman explicitly warns about harvesting “fake ScrumMasters.” Larman’s quote, “Changing the title of someone to ‘Scrum Master’ while he acts like — and is encouraged by the organization to act like — a project manager”⁵ alludes to the fact that simply relabeling old roles into new ones does not cultivate better behaviors and does not improve culture.

Although there are many PMs who are willing and capable of undergoing a mind shift to become SMs, they still represent a fraction. Some PMs also still feel that becoming a SMs is a step down in their careers, a demotion in a way, because now they no longer have the power to control the actions of others. They feel discouraged by the situation and start seeking other career paths. Are their employers aware of that?

What is also worth mentioning is that, unlike the PO (we still refer to Scrum roles for the reasons mentioned above), the ScrumMaster position is rarely a full-time job. Unless the SM is sanctioned to support multiple teams, which is also not always desirable, it remains a part-time facilitator role. Under ideal conditions, the SM role can be picked up by any team member, or, what is even better, by a member of another feature team (this way, a team will be able to completely avoid a conflict of interest and ensure neutrality). Since the SM role is not a full-time job in the majority of cases, it also means that a person who holds it is expected to have specific functional (technical or semi-technical) responsibilities that can benefit a feature team in product development.

This brings the discussion to the following question:

Is a non-technical PM willing and able to gain additional technical skills and functional expertise to remain a valuable asset within an organization that is undergoing Agile transformation?

For many non-technical PMs, learning technology is not an easy task. It is not always easy to switch from many years of using conventional project management tools and techniques, such as a project plans, project charters, WBS, Gantt charts, and individual task assignment lists to Java console, class libraries, Web services API, automatic test tools, or Agile software development collaboration tools.

Often, project managers select less technical direction in their transition. They retrain as BAs and embed with teams, where they begin serving the purpose of PO or PO-proxy conduits, by supporting teams with business requirements (backlog items).

This shift certainly helps in retaining resources, which is always a positive characteristic of any corporate culture, but it only works if there is a very specific need for having a BA embedded with a team (this is more practical in distributed Scrum with offshore teams). Otherwise, creating an extra layer in the business-to-technology communication channel is not recommended. And again, supply-and-demand rules suggest that there are not enough team BA vacancies to accommodate all requalifying PMs.

Epidemic of Certifications

“It is not enough to have knowledge, one must also apply it. It is not enough to have wishes, one must also accomplish.”

-- Johann Wolfgang von Goethe

Problem Statement

Over the last couple of years, the variety of Agile certifications has significantly increased. Specifically, entry-level certifications that attest to basic Agile knowledge (Scrum framework, specifically) are now available in abundance. The type and depth of knowledge that these certifications offer (usually superseded by a condensed training course) are similar and, to a large extent, they cover identical topics.

If we search for reasons why there is such an abundance of basic-level Agile certifications, it will become apparent that it has become more about market share retention by certifying organizations that operate in the Agile arena than about delivering unique, universally acceptable attestation standards that, on one hand, truly reflect individual hands-on practical expertise and theoretical knowledge and, on the other hand, enable individuals to use earned Agile credentials for securing a competitive job.

Discussion

This discussion is not about comparing one certification to another or suggesting which certification is better. It is also not about sharing research data about which certifications are more or less recognized by the industry. Most certainly, the intention is not to compare pricing or promote any particular certification — they are all, when packaged skillfully, attractive. If a reader wants to fully grasp the variety of introductory Agile certifications available today, he can always do a Web search that will produce at least a half-dozen options.

The main purpose of this discussion is to stress the following point: there are *a lot* of comparable certifications to pick from, and this creates unnecessary competition in a space where a universally recognized accreditation standard would be much more desirable.

Competition between organizations that takes the form of “my Agile is better than yours” creates confusion for those who want to get certified and capitalize on their fairly earned practical experience.

It also must be noted that by harvesting so many redundant certifications, the industry creates favorable conditions for “certifications collectors” — individuals who obtain certifica-



tions for the sake of being certified. By doing so, such individuals skew the count accuracy of those who are really qualified for a job and fade the distinction between true practitioners and acronym collectors. At times, we see names of professionals in the Web space where certification abbreviations by far exceed the length of an individual's first and last names, combined. This is truly ironic. As practice shows, holders of multiple redundant certifications have little or even no practical hands-on experience in Agile space.

Finally, there is no statistical proof that would support a claim that any company-employer recognizes only one type of certification over other types. This creates yet another challenge for professionals seeking employment, as they do not know which certification to pursue in order to increase their chances of being hired. If a company is looking for a certification abbreviation next to a name, instead of experience, it will most likely end up with an under-qualified candidate who will further discredit a certification by not being able to live up to a company's expectations of his subject matter expertise.

Conclusion

Let's restate the initial purpose of this discussion. The goal was to make a reader come out of his comfort zone and think about issues that are often omitted from "happy path" Agile themes. The goal was neither to suggest any conclusiveness on the subject matter nor to steer the reader toward any particular actions.

After reviewing this discussion, each reader should be able to develop his own objective perception on a situation, his own independent view and perspective.

Still, how each individual perceives this information will, to a large extent, depend on such factors as:

- Is the reader's perspective personal or organizational in nature?
- How close to Agile transformation activities is the reader positioned today and how soon (if at all) will he be impacted by them?

Each factor by itself is influential, as are both of them in conjunction.

To some, this reading could be a great eye-opener and motivator to make personal adjustments to ensure job security and competitiveness in the job market. This could take on the form of becoming more valuable to one's own organization through pursuing education or training, or by becoming more selective of accreditations and certifications, or by planning an exit strategy to a workplace where conditions for non-Agile activities are favorable.

To others, especially to those whose views are more aligned with organizational views (e.g., high-level executives, C-level officers), this writing could serve as a push toward internal adjustments: reorganization and/or retraining of resources; reconsidering employees and awards and incentives models; revisiting contracts and SLAs with internal and external partners/vendors. By the same token, introducing inspection and validation checkpoints to ensure that there is continuous and gradual advancement toward long-term strategic goals would be another likely outcome of reading these pages.

Further, there will be a certain percentage of readers who will downplay the significance of this writing because of confidence in their "safety zone" away from Agile transformation (e.g., moved to a different department or company) or because they just do not expect a full impact of Agile any time soon (e.g., due to its slow adoption by the organization they work for).

Lastly, there will be parties (varying from individual employees to service providers to client companies) who will react to this discussion with disapproval and defensiveness for self-serving reasons. Most likely, such a reaction would be proportional to their level of involvement with Agile, from the perspective of business development, commerce, and enrichment. Here, fears of becoming a subject of scrutiny and an “ethical audit,” should some of the issues that are being raised here find their way via wider broadcasting channels, will be the main driving force.

Endnotes and References

1. Mironov, Rich. *The Art of Product Management: Lessons from a Silicon Valley Innovator*. Createspace, 2008.
2. Pichler, Roman. *Agile Product Management with Scrum: Creating Products That Customers Love*. Addison-Wesley Professional, 2010.
3. Poppendieck, Mary; Poppendieck, Tom. *Leading Lean Software Development: Results Are Not the Point*. Addison-Wesley Professional, 2009.
4. Cockburn, Alistair. *Agile Software Development: The Cooperative Game* (second edition). Addison-Wesley Professional, 2006.
5. Larman, Craig; Vodde, Bas. *Scaling Lean and Agile Development: Thinking and Organizational Tools for Large-Scale Scrum*. Addison-Wesley Professional, 2008.
6. Logan, Dave; King, John; Fischer-Wright, Halee. *Tribal Leadership*. HarperBusiness; reprint edition, 2011.
7. Reinertsen, Donald G. *The Principles of Product Development Flow: Second-Generation Lean Product Development*. Celeritas Publishing, 2009.
8. Smith, Preston G.; Reinertsen, Donald G. *Developing Products in Half the Time: New Rules, New Tools* (second edition). Wiley, 1997.
9. Mezick, Daniel. *The Culture Game: Tools for the Agile Manager*. FreeStanding Press, 2012.

See more at: <https://www.scrumalliance.org/community/articles/2014/july/unspoken-agile-topics#sthash.730D3h5T.dpuf>

This paper, originally written in February 2013, brings to light some of the least-discussed topics and consequences of “broadband Agilization” that currently take place in the industry. The materials of this paper are subdivided into two general sections:

The first section describes certain impacts that Agile has on individuals and their personal career advancements.

The second section describes organizational-level Agile impacts that pertain more to client companies that undergo Agile transformation, as well as service-providing vendor companies that deliver Agile-transforming expertise to their respective clients.

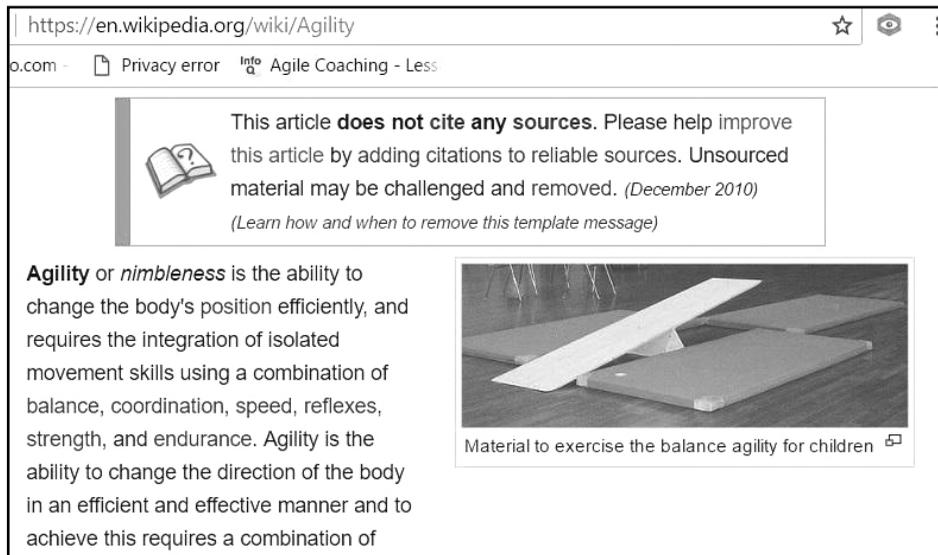
The reader will most likely focus on the section that best represents his primary interests and concerns. However, it is recommended that both sections are read in full, as in unison they create a better holistic perspective of the industry changes brought about by Agile-mania.

The reader will be taken out of his comfort zone and forced to think more uninhibitedly and realistically about those aspects of Agile that may not be as obvious and are not explicitly covered in other literature.

Bad Choice of Verbs Associated with “Agile” by EFL People

Original Source: <http://www.keystepstosuccess.com/2017/06/bad-choice-of-verbs-associated-with-agile-by-efl-people/>

These days, almost everyone knows that organizations cannot “do” Agile; they can “be” Agile. And today, this contrast is used not just by Agile coaches and Scrum Masters.



The screenshot shows a web browser window with the URL <https://en.wikipedia.org/wiki/Agility>. A prominent warning box is overlaid on the page, stating: "This article **does not cite any sources**. Please help improve this article by adding citations to reliable sources. Unsourced material may be challenged and removed. (December 2010) (Learn how and when to remove this template message)". Below the warning, the text of the article is visible, starting with: "Agility or nimbleness is the ability to change the body's position efficiently, and requires the integration of isolated movement skills using a combination of balance, coordination, speed, reflexes, strength, and endurance. Agility is the ability to change the direction of the body in an efficient and effective manner and to achieve this requires a combination of". To the right of the text is an image of a wooden balance beam on a wooden floor, with the caption: "Material to exercise the balance agility for children".

Everyone likes building this fancy figure of speech in their daily lexicon: managers, analysts, developers. Great!!! Below is a snippet from Wikipedia, defining the word “agility“, using the most natural reference: a human body.

From reading the definition, it appears that body agility is equivalent to a body fitness/health. And if so, it would be fair to assume that when we talk about organizational agility, we also talk about organizations, being fit and healthy (organizational fitness/health). Just like a body cannot “do fit or do healthy”, organizations cannot “do fit or do healthy”.

But while wrongfulness of “doing Agile” is mostly admitted today, there are many examples of using other sophisticated synonyms of “doing” that hint to the fact that people are still NOT clear about what Agility is.

As the title of this post suggests, and this is where the biggest irony comes from, the most advanced EFL people (EFL = English First Language) have been making the most noticeable language omissions, while attaching “sophisticated corporate terms-verbs” (other than “do”) to the word “agile”.

Below, is the list of verbs that are not advisable to be used in conjunction with the word “agile”:

- “Implement Agile”
- “Adopt Agile”
- “Use Agile”
- “Introduce Agile”
- “Accept Agile”
- “Follow Agile”
- “Move TO Agile”
- “Transition TO Agile”
- “Transform TO Agile”
- “Install Agile”
- “Administer Agile”
- “Leverage Agile”
- “Upgrade to Agile”
- “Practice Agile”
- “Establish Agile”
- “Experiment Agile”
- “Standardize Agile”
- “Execute Agile”

What is advisable instead: is just to **BE** agile.

Daniel Gullo, CEC, CST



Daniel James Gullo, (CST, CEC, CSP, CSPO, CSM, PMP, ACP) has been a well-known and highly regarded servant of the Agile community for many years. His tireless dedication and effort has earned him the distinction of *The Most Valuable Agile Professional* award for 2015, *The Most Valuable Agile Professional* award for 2016, and *Most Popular Scrum Professional* for 2016.

As CEO of Apple Brook Consulting® (ABC), he has first-hand experience of what it takes to make businesses successful in a highly competitive marketplace.

Daniel is the founder of and chief advisor to Agile Delaware. His experience includes delivering keynote addresses for conferences such as Scrum Gathering – India; Scrum Gathering – Rio; Scrum Gathering – China; et al. He is also co-founder of AgileNEXT – The Next Generation Agile Talk Show and CoreAgility – a monthly publication with Agile articles and other information.

Daniel was Conference Chair for the 2015 Scrum Gathering in Phoenix. He was also the Conference Chair for the 2013 Scrum Gathering in Las Vegas, which makes him the only individual in the community to serve twice in this capacity.

Daniel has served on the Trainer Acceptance Committee (TAC) for the Certified Scrum Trainer (CST) certification program. He has also been a reviewer of Certified Enterprise Coach (CEC) applications. Daniel was a founding member of the Scrum Coaching Retreat Planning Committee; host of Coaches Clinic events; and facilitator of Open Space Events, and numerous other activities.

As an Impact Partner and Investor for Fresco Capital, Daniel is passionate about enabling small companies with big ideas to live up to their full potential and achieve success. Daniel also serves as a mentor for The Laudato Si' Challenge, a Vatican-sponsored startup accelerator program inspired by Pope Francis.

A Service-Disabled Veteran-Owned Small Business (SDVOSB), Apple Brook Consulting has donated over \$75,000 (as of June 2017) to active-duty, veterans, and retired military members through its discount program on training and via other methods.

Daniel's book, *Real World Agility: Practical Guidance for Agile Practitioners* includes practical, real-world answers to practical, real world questions from students and clients. It is available for purchase on Amazon.com and Barnes and Noble.

Daniel Gullo, CEC, CST

Daniel is also co-host and co-founder of AgileNEXT: The NEXT Generation Agile Talk Show, which is available on SoundCloud, iTunes and at: <http://www.agilenext.tv>

Website

<http://www.apple-brook.com>

Email

daniel.gullo@apple-brook.com

LinkedIn

<https://www.linkedin.com/in/danielgullo>

Twitter

[danielgullo](#)

Business Phone

(+1) 484-857-2786

5 Years Ago . . .

Original Source: <https://www.linkedin.com/pulse/five-years-ago-daniel-gullo>

In 2009, I was hired by Project Management Institute (PMI) as an Agile Coach. I was very excited and felt like Obi Wan Kenobi venturing into the Death Star to blow it up internally with disruption and a world of possibilities . . .

As many people are aware, PMI generally represents waterfall project management, command and control, and prescriptive methods. Agile, on the other hand, stands for flexibility, change, learning, openness, innovation, empowerment, customer delight. Though I was also a PMP at the time (and still today), I had come to the realization that the PMBOK Guide style of project management was becoming more and more of a dead man walking in light of the myriad changes in technology that happen daily.

My responsibilities included helping the organization shift from a waterfall mindset to Agile by using Scrum practices. MyPMI.org had been through a few different botched vendor implementations and was in desperate need of redesign, having been built on top of the SharePoint platform (not really intended for a > 500k person, public site application).

Leading up to this role, I had coached several other major organizations through Agile transformation and PMI was fairly impressed with my accomplishments on these consulting engagements.

While I was working at PMI, I led the creation of an Agile certification called PMI-ACP (Agile Certified Professional) as the Project Manager assigned to the effort by PMI. This effort put me in touch with many of the Agile “luminaries” and thought leaders, with whom I became very good friends, sought mentorship, and so on.

In addition, I was beginning to attend numerous Agile events and meet many other thought-leaders and senior level Agile coaches and trainers. Specifically, I was becoming good friends with many CSCs and CSTs (Certified Scrum Coaches and Certified Scrum Trainers) who were generally seen by the Agile community as the “rock stars” of the field.

In fact, prior to meeting and getting to know these individuals personally, I looked upon them in a similar way as the gods of Mt Olympus. As I became increasingly better acquainted with them, I began to envision myself as a “rock star” as well; i.e. part of the club, though I was neither a CSC or CST. I had begun to give talks at various Agile events and was beginning to build a solid personal brand.

In 2010, I decided to apply for CSC myself. With 4 solid years of Agile coaching and training experience in Fortune 100, Fortune 500, and other organizations; counting all the key players as good friends; and my reputation due to involvement volunteering in the Agile community, I reasoned that I would have a very easy time with my application.

In fact, I was so confident in my abilities, that I didn't really take the application seriously. I

spent about 4-5 weeks putting together about 10 pages of content in response to the questions that were asked.

By the time that I submitted my application, I had moved on to be a head Agile Coach for NAVTEQ. I hired Bob Sarni, a CST and CSC, to deliver certified training on behalf of my client (NAVTEQ). Shortly after, Bob also ended up being the team lead for the reviewers who processed my CSC application. Again, I thought that I would be a shoe-in since I had always heard about the importance of making connections and having a strong network, Bob knew my abilities, and so on.

I failed.

Not just a little . . . completely. Utterly. Miserably.

In retrospect, my answers were atrocious. In some instances, I had grossly disregarded the instructions for the questions; e.g. on one question, it stated “give three reasons . . .” I gave two. My answers were shallow, cursory, and incomplete where they had asked for depth and breadth.

Bob and I made an appointment to review my application. As he patiently walked through the application with me on the phone, tears began to well up in my eyes. I was so ashamed. Not just hearing about how poorly I had done on the application itself, but the full realization of how pompous and arrogant I had been hit me at that very moment. It was clear that I had been acting very entitled, like a spoiled child who deserves to have something that they have not earned.

I thanked Bob for his time and his valuable feedback. I also apologized to him and asked him to relay this message to the other review team members due to the fact that I had wasted their time.

I was angry . . . with myself.

I spent some quality time reflecting on my folly before I began working on revisions to my original application. I vowed to work on these revisions every spare moment I had when I wasn't working or taking care of familial duties.

Over the next 4 months, I drafted a response to the application questions that was almost 3 times as long. The updated copy included deep reflection, painful attention to detail, and many, MANY iterations reviewing, revising, etc. I talked about how my background in law and psychology had influenced my coaching style and attempted to paint the picture of how I had been successful (and not) with clients such as IRS, NAVTEQ, PMI, VWR International, Invista, Credit Acceptance Corp, T. Rowe Price, and others.

I finally resubmitted my application and breathed a sigh of relief. I felt hopeful this time; not smug or overly confident. I still had some degree of doubt in my mind “What if I missed something?? What if I am wrong? What if I didn't articulate something clearly?” The previous experience had taught me a valuable lesson about self-reflection and self-awareness. I ruminated on this night and day for several days and finally resolved to relegate any thoughts of inferiority or self-doubt that I had to Imposter Syndrome. Overall, I was feeling that this time, I would be successful.

I waited . . . and waited . . . and waited.

It took much longer this time to hear anything back from the review team. That was start-

ing to irritate me and I thought “The mark of a good coach is patience. Assume positive intent and moreover, consider the perspective of those involved; with the Scrum Alliance, the review team, and so on.”

In fact, I reached out a couple times to see what was going on. I didn’t want to be pushy or rude. However, with each passing week and no response, I began to grow increasingly more anxious and impatient. The self-doubt and angst were ever present . . .

Finally, I heard back from the new leader of a new review team. This was Martin Kearns, a CSC and CST. I had never met Martin before and knew almost nothing about him other than he was located in Australia. Martin and I scheduled a Skype call to discuss my application.

At this time, I was working as an Agile coach for the Internal Revenue Service (IRS). On the day of our meeting, there was a very large earthquake in Virginia which caused all rail traffic to be halted. I was stranded in New Carrollton, MD and was lucky to find a hotel for the evening. It was not the ideal environment or mindset for a discussion about my CSC application. Martin suggested rescheduling the call. However, I dismissed this suggestion because I was more anxious to hear about the results of my CSC application than about the earthquake that had occurred. I thought for certain that Martin would be bringing me glad tidings of great joy by congratulating me on passing the review and also, sharing some feedback about my application that I could learn from.

I was wrong.

Martin had the unfortunate task of informing me of yet another failure on the CSC application! This time, I was angry at the process and the review team, not myself. I was in a very reactive mindset. I wouldn’t say that I felt entitled, but there was a hint of that which was interfering with my ability to objectively process what Martin was telling me. In my mind, I was thinking “I have spent all this time, working on this, refining it, tempering it, having others review and provide me feedback . . . and you are telling me I failed??? How dare you.”

As Martin and I chatted about my answers to the questions, I provided some clarification and he gave me feedback. I really tried not to get defensive, but rather, provide some color and context with the hope that maybe THAT would help him understand and maybe the team would give me a pass; some understanding, compassion, and so on.

At one point I asked him how far I was from making it this time. He hesitated. That was one of the longest, most uncomfortable pauses that I can recall in any conversation I have ever had. Finally, he spoke.

I had done WORSE this time than last time!

He mentioned to me that there was an internal scoring system that the review teams were using and that my scores from this team on the application were actually lower than on my first attempt.

How could this be? This response defied logic. How could I possibly do worse when I put my heart and soul into my responses on the application questions than the time when I took a completely cavalier attitude toward the application? I certainly knew how to coach organizations; leadership to individual contributor, top-down, bottom-up, middle out, thinking in terms of complex adaptive systems, etc. I had a great deal of experience and I was constantly looking for ways to grow and learn. I felt that this time, the system must have failed me

instead of me failing the system.

In fact, at this point, I recall that my attitude soured and I started to become confrontational with Martin. I asked him about this “elusive and secret scoring system” that the applicants are not aware of. I asked him if he could share the criteria so that I could have an understanding of what I was being evaluated on. However, he would not tell me anything more than there was a system of scoring the applications. I was skeptical and began to internalize this decision . . . “Someone on the inside doesn’t like me. They have it out for me. That’s the only possible explanation. I pissed someone off and now I am paying for it.”

I figured that I had nothing to lose at this point, so, I resolved to write a very detailed letter giving my feedback on the application process and send it to the head of the program, Pete Behrens. I did try VERY hard to be objective and level-headed in my communication. I tried to bring possible solutions to the issues that I was raising. Again, I thought “Maybe the organization with the certification for coaches needs some coaching? I will try to open a door for that with my Email.”

Pete asked if we could discuss on the phone. He was very kind, understanding, and proposed that maybe there were some disconnects between what I thought the questions were asking and what the reviewers were expecting in terms of answers. My analysis of the CSC process and my reasoning, etc., demonstrated to Pete that I indeed was worthy to be a CSC, or at least had very strong potential.

I agreed that there were probably some significant disconnects on both sides. My writing style tends to be a bit more casual and even colloquial. Maybe that was the issue? That’s how I feel most comfortable in expressing myself.

I shared with Pete that I had no desire to be a CSC any longer; mostly because I couldn’t handle going through yet another round of application revisions that took months and months only to face a review team (and yet another possible rejection). I was transparent and shared with him that I was bitter about the process, Scrum Alliance, and I also mentioned that I didn’t want to be a part of the community if it was so inflexible with its requirements that it couldn’t take into account the value of an individual beyond what their application stated.

Pete mentioned that he would be willing to mentor me for a period of time until we mutually felt that I was ready to be granted the certification. It would be an experiment. I really liked that idea. Coaching is all about the human element, a relationship, getting to know one another. In fact, I thought “Why couldn’t this be THE process? You know, coaching someone along the path of becoming a certified coach?? A candidate demonstrates a particular level of mastery and competence in coaching and then they are coached / mentored the rest of the way . . .”

And so, Pete and I embarked on a 9-month mentorship; meeting monthly to do a retrospective on the previous month’s learning, plan out the next month’s learning, and discuss any impediments. I was also frequently reaching out to Bob, Martin, and many others to learn from them since they were becoming trusted advisors and close friends. They helped me to grow and really hone my coaching stance. We exchanged ideas and resources; talking about Virginia Satir, Gerald Weinberg, Bill Joiner, Peter Senge, Marshall Rosenberg, Harrison Owen, and many others who had influenced and inspired us.

One day, Pete asked me if I was ready to become a CSC. Happily, I said that I was. That was five years ago this month (June 2012).

This entire experience was life changing for me.

When I was finally confirmed as a CSC, I felt like I had REALLY earned it, yet, it was like removing the veil and discovering that there is a long way to go in my learning journey. It wasn't like sticking a quarter in the gumball machine and out comes the CSC certification. I was grateful for the struggle, the pain, the challenge. I had matured quite a bit; in humility, understanding, emotional intelligence, awareness. I began to explore mindfulness practices with some degree of success and incorporated that into my coaching approach.

I learned that sometimes it's not just about what you know but also whether you realize how little you know; recognizing the conscious and unconscious competence, acknowledging the conscious incompetence, but also considering the unconscious incompetence and speculating on how to address that gap. Also, leaving an opening so that you may be receptive when someone else points out your unconscious incompetence.

I cannot thank Bob, Martin, and Pete enough for all they did for me during the time that I was going through the CSC process and in the years since. There are also many others who I am deeply grateful for because they have influenced me deeply with their insights and willingness to discuss, respectfully: Roger Brown, Dhaval Panchal, Lyssa Adkins, Peter Hundermark, Jürgen Hoffmann, Andreas Schliep, Sabine Canditt, Dan LeFebvre, just to name a few.

In the end, Scrum Alliance did change the process for becoming a CSC (now called CEC – Certified Enterprise Coach) to include earlier and more frequent feedback and a coaching / mentoring element. I have spent considerable time mentoring others who, like me, are feeling their awkwardness and struggling with that on their journey to become better coaches. I have served on the review team for CEC.

Recently, someone shared that the main driving factor for why Scrum Alliance changed the process — the main catalyst to effect this significant change — was the experience that I had coming through the program and the experiment that Pete and I conducted together connecting on a human level to understand each other and fitness for the program on a more intimate level.

Reflecting on the events surrounding my certification as an Enterprise Coach in support of my PhD application has been a fantastic exercise! I am happy for this element in the application because it gives me a concrete reason to do zoom back in with a lens of introspection and consider my journey as I zoom back out. More importantly, this essay assignment has confirmed for me that I am ready to tackle the next leg of my learning journey by discovering new ideas, relationships, and opportunities that the program will afford me.

Peace and blessings.

Make Agile Great Again

Original Source: <https://apple-brook.com/1473501-2/>

Over the last 2+ years, I have seen the topic “scaling agility” become the hottest new buzzword and market trend.

Everyone from executive leadership to people on development teams are wondering: “How do we scale?”

What they really want to know is: “How can we eat all the cake, ice cream, pizza, and other garbage we want, never go to the gym, and actually LOSE weight???”

They want to know how to improve their organizations without making a single trade-off.

They want absolute certainty that what they are doing will yield a certain return on their investment BEFORE they make the investment.

Time and again I advise organizations that if they want to improve, they need to begin with making smaller investments and conducting shorter experiments. They seem to be willing to do “anything”, except . . .



Have Dedicated Scrum Masters for EACH Team

We have thousands of professional sports teams in the world who are the best athletes in their sport. They know the game, the rules, how to win. Yet, every team has coaches . . . in fact, some teams have several coaches for ONE team.

If we want to have a high-performing Scrum Team, then we need a dedicated, full-time coach for each team whose sole job is to ensure the team has everything they need, looks for ways to improve, resolves issues, etc. and who doesn't have the conflict of interest represented by coaching multiple teams simultaneously.

THAT is the ScrumMaster.

I have seen countless organizations who try to cheap out by having a Scrum Master “manage multiple teams”.

FACEPALM!

“You just don't get it . . .”



Then, they blame Scrum because they aren't getting the increased Velocity that they expected . . .

DOUBLE FACEPALM!

I have also seen some organizations who take the Scrum Master role very seriously and those organizations have slowly but surely improved over time.

Have Dedicated Development Team Members For EACH Team

"QA is a bottleneck." said every organization I have ever worked with who has 5 Developers writing code which ONE person tests . . .

"We are stuck. No one from the [DBA / DevOps / Documentation / UX-UI / etc.] Group is available right now." said every organization who has so-called "shared resources". (Calling people "resources" is offensive, btw.)

When Development Teams lack a critical skill set, there is bound to be trouble.

Imagine an military unit that doesn't have any medical capabilities. "We don't have wounded people all the time. We can't afford to have someone who is dedicated to rendering aid for each unit."

It's ridiculous, right??

Instead, why not invest in training so that EVERYONE on the team knows enough about [database / documentation / UI / testing / etc.] to get the work done? If there is still the need for an SME, then hire that person... and train them in other skills so that they can do other stuff when they aren't performing in their speciality.



It's just common sense, folks.

In fact, I think the entire Agile Manifesto could be simplified down to one statement:

USE COMMON SENSE

However . . .

"Common sense is not so common."

– Voltaire

Have Dedicated Product Owners

People love to bring up edge cases and exceptions as if they were the rule.

I am often asked: "Can someone be the Product Owner for numerous smaller projects?"

I usually answer with a question: "So, these smaller projects aren't really important?"

Sure, if there are a bunch of small systems/applications that are very far along in maturity to the point where they don't require any real significant changes and the systems/applications themselves are not critical to the mission of the organization, then, I suppose a single person can be the caretaker of those decisions and modifications.



However, when this answer is given, the logic is applied to “projects” that ARE main lines of business for the organization. I see Product Owners who are responsible for 4-5 different mission critical products and then leadership wonders why these products are failing.

“Is it any surprise that the products are 20% successful when you have a single person acting as the Product Owner for 5 products??”

The Product Owner is the most misunderstood role in Scrum. I haven’t seen many organizations who truly get it. It’s a tough role to play, in all honesty. The person needs to be talking to customers and stakeholders daily. They need to be using the outcomes from those discussions and using the information to refine the Product Backlog. They need to be learning how to define small increments of value, which are elements of the solution a customer is looking for; aka Features. They are monitoring the financial health of the product; answering questions that the Development Team has; and even verifying that the product is accepted throughout the Sprint.

There is plenty of work for 2-3 people to do, if the role is properly understood. I seldom find products that are flourishing where the Product Owner is responsible for other products as well.

If the organization makes the investment in having a full-time champion for the customer and product, then it is bound to be successful.

Everyone Knows

I am trying very hard to spare you from my Leonard Cohen impersonation here . . .

Everybody knows what it takes to be successful.

Dedication. Sacrifice. Trade-offs.

The goal is not to see velocity increase. The goal is not to aim for a certain ROI that a product is expected to make. The goal is to balance delighting the customer with what makes sense for the organization.

That requires constant conversation, experimentation, learning, etc.

“Aim small, miss small.”

-unknown

That is, focus on making things work great at the team level before you start worrying about how to “scale” patterns of dysfunction.

The Lost Art of Management

Original Source: <https://apple-brook.com/the-lost-art-of-management/>

When I ask most people why they wanted to become a manager, the answer is almost invariably “more money.”

This answer is consistent with my own motivations as I think back many years ago in my own career path.

I would also say that it was a necessary stepping stone within some organizations in order to move higher up in the organization.

Organizations that are more progressive and forward thinking may still use the word “management” but in reality, they focus on “leadership” skills and cultivating an environment of collaboration, inclusion, ownership, teamwork.

Most of the education I have had over the years has been focused on Leadership. I have read many great books that I like to recommend to people: *Radical Management* – Steve Denning; *Moments of Truth* – Jan Carlzon; *Leading At The Edge* – Dennis Perkins; *Leadership Agility* – Bill Joiner, and now *Extreme Ownership* – Jocko Willink. There are SO many others.

In helping organizations become more innovative and focused on customer delight, I find that the activities of management don't really go away.

There is a transference.

People are expected to act with maturity and accountability at ALL levels of the organization. There is trust. Organizations pay people a fair amount and don't resort to stick and carrot tactics to entice people or compel people to do what they want them to do.



Our goal is to help people regain their sense of intrinsic motivation that has been lost through years and years of systematic abuse in toxic command and control corporate environments.

Much has been written on the topic of motivation by Daniel Pink in his book *Drive* and Alfie Kohn in his books, including *Punished By Rewards*.

Threats do not work.

As a knowledge worker, if I am being threatened or working in a toxic environment, I won't tolerate that for very long. I am smart enough to realize that I have other options available

to me. I can see the massive volume of job postings and opportunities out there. My inbox is continually flooded with Emails from recruiters. If someone is mistreating me, I will be gone when I find the first opportunity to leave. Or, better yet, I will just go start my own business!!

Bonuses do not work.

If my base pay is 80% of my total compensation package with a 20% bonus, it's not like I with-hold 20% of my brain power until I receive the bonus. As a knowledge worker, I am putting in 100% of my effort all the time. The only thing a bonus can buy is more of my time. However, that is problematic. We know that when people work more than about 8 hours / day, their performance and quality of work degrades. So, if I am working 12-16 hours per day, yes, the company is getting MORE of my time, but what I am producing is essentially crap.



What does work?

Figuring out what is acceptable as a guaranteed salary.

It takes money off the table and allows both the organization and the employee to focus on producing value instead of playing games to earn the bonus or worrying about whether the bonus will be paid.

Managers are not necessary.

Managers are a product of Taylor era factory work, not knowledge work. Taylor's research, while revolutionary and meaningful for its time, is now outdated. Yet, our model of organizations has not changed much in the last 100+ years.

For highly predictable work, it makes sense to have less variance in production. However, for the type of work that we do in software — innovative, constantly changing, complex work — more thought and variance is needed. Collaboration, discussion, deviation from the norm, and experimentation is necessary.

Many lose sight of the fact that Scrum is empirical rather than predictive process control.

Often times, I hear of “resource constraints” for skills such as DBA, UX/UI design, and so on. Yet, there is no shortage of managers. It reminds me of this classic picture from a Dr. Seuss book that I augmented and tweeted out several years ago:

Basically, we need more people DOING and less people telling people “DO!”

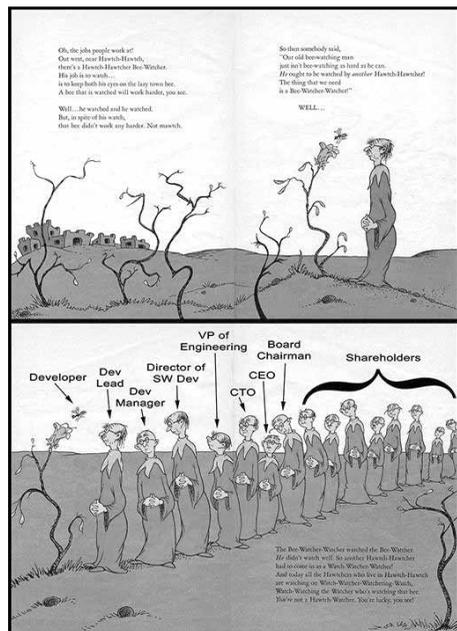
Managers are probably freaking out at this point . . .

Relax.

You are smart people. Accomplished. Driven.

There is hope.

You can refocus your efforts on coaching, mentoring, helping, serving, consulting, and so on. Build communities of practice and then shepherd those. Advocate for training dollars to help people become cross-functional.



Drop back down to the team level and become a contributor again. Become more involved in strategic thinking. Move over to the business side and learn the key concerns related to sales and marketing so that you can represent the customer more effectively; e.g., as a Product Owner.

Step away from the problems and issues and allow the team to step into that gap. Empowerment doesn't happen by waving a magic empowerment wand or making proclamations. Rather, it happens when someone who has always made the decisions stops making them and trusts others to make the decision instead.

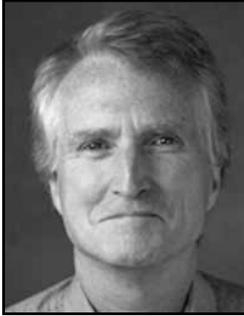
It's a very unnerving feeling, not being in control anymore.

There are many feelings we go through, many fears. However, in time, you will find that people still value your guidance and will actually seek your input more if you are open and approachable.

There is no "manager" per se in Agile. There is unlimited potential for "leadership" however. Focus on growing those skills and characteristics and be less concerned with a specific role or title. You will be amazed at where that path leads.



M. Kelley Harris, CST



M. Kelley Harris is an Agile/Scrum coach, trainer, and developer. He has 20+ years of software development experience, in roles including software engineer, architect, manager, director, consultant, trainer, and coach. He helps innovative teams keep the process simple and essential, and get on with thrilling customers. He draws on Agile, Scrum, Design Thinking, Lean Startup, and more. He brings a spirit of help, compassion, service, collaboration, and innovation. As a Certified Scrum Trainer (CST), he has trained and coached people in the U.S., Canada, Europe, and India.

Kelley has developed software in the domains of scientific instrumentation, nanotechnology, semiconductors, solar energy, automotive diagnostics, investment, real estate, music, eCommerce, etc. Since embracing Extreme Programming (XP) in 2002, and Scrum in 2004, he has helped a wide variety of teams utilize Agile principles. He has been fortunate to have worked with Agile thought leaders including Ward Cunningham, Robert Martin, Joshua Kerievsky, and more. He founded SourceCell in 2004, and currently does coaching & training. He attended the University of California (B.S. Physics; B.A. Economics-Math). He holds three software patents in nanotechnology. He is based in Palo Alto, California (Silicon Valley) and Santa Barbara, California (Silicon Beach).

Innovation Trends to Befriend: Empathy, Safety, Culture and Invitation

Original Source: <http://sourcecell.com/blog/2016/01/02/innovation-trends-to-befriend-empathy-safety-culture-and-invitation/>

How can we all get better at innovation in a hurry? I attended a bunch of conferences and workshops this last year on Agile, Lean Startup, Design Thinking, etc., and noticed some simple powerful trends in the conversations. You've likely already heard of these trends. What is new is their rapidly growing support from diverse communities, singing the same tune.

Empathy – Talk to Real Customers

Really talk (and listen) directly to users, early and often. This is the chorus coming from Design Thinking, Lean Startup, Lean UX, and Agile thought leaders. It pulls from “human-centered design” or “human-centered innovation.” Why aren't we doing it more? It does take time and humility. As a start, we can rely less on intermediaries and intermediary means (e.g., “user stories”), and “voice of the customer” initiatives, and get the makers actually hearing real stories from real users. We can embrace the proven insights of Design Thinking's focus on developing user empathy and understanding through direct interviews and iterative prototyping. We can use the insights from Lean Startup to validate that we're building the right things, with the right business model.

Psychological Safety

If we can reduce the fear of judgment, creativity will happen naturally as practiced by David Kelley and the Stanford Design School team (famous for Design Thinking). Google studied the attributes of their effective teams and found that “psychological safety” is the most important attribute by far. Wow. Let that sink in. It wasn't skills. It wasn't tools. It was “psychological safety.” Agilist Joshua Kerievsky has been promoting broad safety for numerous years. In James Tamm's book, *Radical Collaboration*, Tamm finds we will not get collaborative behavior into all our organizations unless we reduce the fear, blame, etc., that inhibit collaboration.

Create a Culture Where Innovation is Inevitable

Peter Drucker is often credited with saying, “Culture eats process for breakfast.” Master gardeners stop focusing on feeding specific plants and instead focus on “feeding the soil” knowing the soil will support the plants. They experiment. They learn. We can do that in organizations, by encouraging experimentation, collaboration, and learning. Management's job can move away from selecting and managing projects, and towards creating environments and cultures that create great products and services.

“The object isn't to make art, it's to be in that wonderful state which makes art inevitable.”

Robert Henri.

Invitation Over Policing

Policing people & processes seldom produces the best results. There is a growing move towards inviting people voluntarily into change initiatives and giving them lots of choices. One of the key Agile values from the Agile Manifesto is Individuals & Interactions over Processes & Tools. Self-organization can start early, and grow deep and wide. Fredric Laloux's book *Reinventing Organizations* contains beautiful examples of companies that have embraced this and produced amazing results.

The good news and bad news is these are largely issues of mindset, choice, and emphasis. We could change fast if the conditions and motivations are right. Other teams may need to be rebuilt.

While big organizational change would be dramatic, I believe we can all innovate more now by getting to know our users, reducing fear, creating a culture of experimentation, and giving people choices.

Jon Jorgensen, CTC



Jon Jorgensen is a Certified Team Coach with the Scrum Alliance and an Accredited Kanban Trainer with Lean Kanban University. He takes a holistic approach to assisting large and small organizations successfully navigate their journey through Agile transformation.

Working at a global scale with multinational conglomerates such as Canon, EMC, TiVo/Rovi, and Adobe, Jon has pioneered the use of Open Space Agility and Enterprise Scrum to support every aspect of an organization, including the culture, in shifting the direction of

its growth to something scaleable and conducive to an Agile mindset.

The only thing that scales is self-organization. As a Certified Professional Facilitator, Jon provides client organizations with many options for a substrate upon which rapid, unfettered, disruptive growth reliably springs.

In coaching alliances with business owners, executives, managers and team members, Jon consistently sees bold leaders emerge newly present to their own greatness and expanded capability to orchestrate big room events wherein large numbers of Agile teams reinvent their processes, align goals, and set work cadence. He incorporates Applied Improv principles into his work with innovation teams and executive leadership.

Jon presents on these topics in industry events including Agile Cincinnati. He founded Agile Coach Camp US West (in 2015), Agile Open Omaha, and Agile Open Phoenix (in 2018) and organizes Agile Meetup Groups in Los Angeles and Omaha.

Follow his tweets [@waterscrumban](#) or just reach out to Jon.Jorgensen@needle-hop.com to explore what it would take for your organization to go beyond just moving the needle.

16 Questions for Setting-Up a Coaching Alliance

Original Source: <https://needlehop.com/16-questions-setting-coaching-alliance/>

As I have begun building a coaching alliance with the CEO and executive team of my clients in the past, I have found a very broad disparity of awareness, maturity of practice and level of depth in coaching needs. So, I'd like to share some questions that I may begin conversations with whoever may be engaging my services. This kind of **Fierce Leadership**¹ conversation (please see the book by the same title) is most useful when it is completed earlier than later in a business transformation consulting engagement.

1. Does the Enterprise Agile Coach's activities include executive coaching?
2. Who does coach the executives?
3. How do the executives shift their mindset to become Agile?
4. In what way do the executives go first in Agile, to serve their constituents?
5. How do the executives continuously and publicly express their sense of passion and responsibility about the Agile transformation?
6. How does each executive visually and transparently communicate to the workforce which structural impediments she is accountable for removing, has removed, won't remove and can't remove? (Do they commit to reading and responding in writing to the **Open Space Proceedings**² within 48 hours of publication?)
7. How does the executive know how effective they are in their attempts to inspire, support, guide, empower and collaborate with and innovate with the Agile workforce?
8. Jay W. Forrester says the Policies determine the behavior of the system. How aware is the executive about the dynamics currently influencing the system, and how frequently do executives revise/update the work system, organizational structure or culture?
9. How do executives continuously improve themselves and stay engaged in lifelong learning?
10. How does the executive demonstrate to the Agile workforce the value and positive career impact of heightened engagement in lifelong learning inside the organization?
11. How active a role does the executive play in the Agile Community of Practice?
12. What are the aspiration goals of the executive to reach their next level in leadership agility?
13. Exactly which of the **6 Types of Agile Coach**³ do you hope I will be? What are you willing to commit to, in service to my fulfilling on the role of Enterprise Agile Coach, to enable me to accomplish this level of coaching?
14. What's at stake in this transformation? How large are the potential upsides to its success, and down-sides to failure?

15. What makes you think you need to go outside the company to get coaching for this?
16. How long do you think it may take for a truly exceptional Enterprise Agile Coach to mentor up his replacement?

Links

1. <https://www.amazon.com/gp/product/038552904X?ref%5F=smi%5Fwww%5Frc%5Fgo%5Fsmi%5Fg2609328962&%2AVersion%2A=1&%2Aentries%2A=0&ie=UTF8&pldnSite=1>
2. <http://openspaceagility.com/big-picture/ost1-proceedings/>
3. <https://www.scrum.org/resources/blog/6-kinds-agile-coaches-which-are-you-hiring-or-applying>

Mark Levison, CST



Mark Levison is the founder of Agile Pain Relief Consulting, a management consulting and training firm. He brings extensive Agile experience to the table, practicing since 2001, and specializes in building High Performance Teams and Organizations.

Mark is a Certified Scrum Trainer whose credentials include over 25 years of IT experience, including positions of Developer and Development Manager. Having experienced what can go wrong when building software, he turned to Agile for improved methods and tools. In addition to Scrum, Mark studies behavioral psychology and neuroscience for a deeper understanding of how teams work.

He is currently writing a book, *Beyond Scrum Building High Performing Teams and Organizations*. An early draft is available at: <https://agilepainrelief.com/notesfromatooluser/2017/01/beyond-scrum-blog-series.html>

“Because Our Competitors Are” is No Reason to Become an Agile Organization

Original Source: <https://agilepainrelief.com/notesfromatooluser/2016/02/because-our-competitors-are-is-no-reason-to-become-an-agile-organization.html>

Companies are starting to fall into a trap, and it goes something like this: “Our partners/competitors/ . . . are Agile, so we need to be Agile.” Becoming Agile without a valid reason will harm your organization. I can’t state that any simpler.

In the ‘80s and ‘90s, rival manufacturers often visited Toyota plants, and Toyota was delighted to welcome them because Toyota understood that even if their competition copied company practices, practices change. What competitors weren’t copying was the culture that created the practices in the first place, and that’s where the real value was. Thus, we had Cargo Cult Lean at many North American manufacturers and it didn’t produce the results that companies were hoping for.

What competitors weren’t copying was the culture that created the practices in the first place, and that’s where the real value was.

We’re seeing that again with Agile. It’s not enough to become Agile and to copy Agile practices simply because your competitors are. Unless you develop a culture that creates its own practices, this will lead only to Cargo Cult Agile, and not a true Agile Organization.

So what are some valid reasons to become Agile? One of the primary reasons is to be able to readily adapt to a changing environment. Other reasons include: Resilience; Predictability; Risk Reduction; Morale and Retention; Early ROI; Customer Satisfaction and Simplicity.

Consider how Blockbuster adapted to online video, or how the taxi industry is responding to Uber/Lyft. Control-focused organizations struggle to adapt rapidly enough to survive, compared to their competitors who embrace change. Traditional/hierarchical organizations work well when problems are clear and solutions are repeatable but, unfortunately, those that thrive in those conditions are fragile when the situation changes and they can’t adapt.

The structure of traditional organizations is one which evolved in a world where the pace of innovation and change was much slower. Changes could be spotted years out, and a response could be crafted and the organization would do well.

*That world no longer exists. Whole industries die in only few years if their response to change isn’t rapid and flexible. Which is where **Cynefin** can come in to the conversation.*

Cynefin

Cynefin is a way of understanding the problem domains in which we find ourselves, and identifying which tools would be appropriate in response.

At first blush it can look a little daunting, but bear with me and you’ll see that it doesn’t have to be. It’s merely a matter of “cause and effect”, and how simple or complicated that plays

out within the context of your organization.

The Cynefin Framework breaks down into 5 different domains to describe different types of relationships between cause and effect – from straight-forward, to complicated, to non-existent.

The first of these is, well, Obvious.

Obvious (formerly called Simple) is just as the name suggests. These contexts are clear to all involved. If X, then Y. At any stage in a process, it is clear what the next steps are. Examples are aspects of banking (interest calculations), insurance (calculation of premiums), etc., Organizations in this realm gain value from a degree of structure to ensure that people follow the rules. Standard practices apply here.

A simpler example? Let's talk about growing things. Develop your green thumb. In an Obvious system, we have a sponge. If you apply water to it, it will swell and grow. Cause and Effect in its simplest form. Sponge + Water = Bigger Sponge.

That's Obvious. The next domain is a little less so.

Complicated is where the relationship between cause and effect is harder to see. It requires analysis and, often, expertise to find and understand the relationship(s). Once understood, "best practices" can be developed in this domain.

To return to the growing analogy, this is the system where we're growing one plant in a pot, which is considerably more complicated than growing a sponge, but there is still logical cause and effect involved. Seed + Soil + Water + Sun + Food = Plant. In this world, there are rules of thumb (green or otherwise) and best practices.

The next domain is more **Complex** and, in these contexts, cause and effect relationships are only understood in hindsight. Given the unpredictability of this domain, we're better off probing, sensing and responding instead of trying to control or plan. Instead of looking for complex solutions, seek simple rules or heuristics to help work well in this environment.

Trying to help grow our kids is an example of the **Complex Domain**. We ask them to do something (probe), they respond (sense) in an unexpected way. It's only in retrospect that we can see why they responded that way. Next time, we adapt (our response), changing the phrasing/tone based on our new understanding of them.

Complex domains require many diverse viewpoints to help solve. Although challenging, they're still much easier to navigate than Chaotic.

In **Chaotic Domains**, there is no relationship between cause and effect. Emergency/disasters are examples of the Chaotic domain. In these cases, the goal is simply to try and bring them back from Chaos to the world of the merely Complex. We don't often see this domain in the business world, so we're not exploring it here.

And the fifth and final domain is **Disorder**, which exists when it hasn't yet been determined what the cause and effect relationship is. It's in this state that people are most apt to make decisions based on their own comfort zone.

These days, businesses are usually working in Complex

It's important to note that one of the risks of a Complicated Domain is that we listen only to the experts. It's vital that we also factor in our own observations and environment.

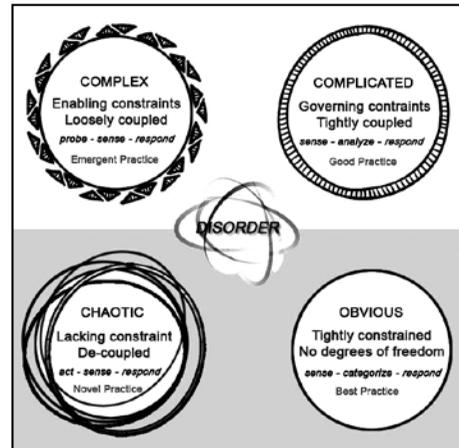
Domains, which means that traditional, old-school approaches aren't realistic.

We can't afford to have the DNA of **Simple** organizations persist if organizations want to thrive in this new, rapidly-evolving world. So we need to create organizations that can adapt — and even thrive — in a **Complex** world, and help **Simple** and **Complicated** structure businesses evolve to that goal. We can do that by creating Agile Organizations that understand the Cynefin Framework, and how different responses are appropriate for different complexities of situations.

Agile Organizations:

- Can sense their playing field;
- Can adapt;
- Have resilience built in;
- Focus on quality;
- Delight the customer;
- Get earlier ROI;
- Target delivery, not risk reduction;
- Build simpler systems and products;
- Create the unimagined;
- Ensure alignment toward a common goal.

Become an Agile Organization because it helps your organization to thrive in a Complex world – not because your competition is doing it. Choose this path knowing that there is a great deal of change involved, but understanding that the change will help create more value in the long run.



References:

- <http://www.anecdote.com/2009/04/a-simple-explanation-cynefin-framework/>
- <http://www.scrumsense.com/blog/cynefin-framework/>
- <https://www.mindtools.com/pages/article/cynefin-framework.htm>
- <http://www.leadingagile.com/2011/01/the-12-key-reasons-companies-adopt-agile/>

Image by *Agile Pain Relief Consulting*. Image elements designed by *Freepik* (http://www.freepik.com/free-vector/calligraphic-circles_762241.htm)

Simplicity

Original Source: <https://agilepainrelief.com/notesfromatooluser/2016/03/simplicity.html>

In 2008, the global financial markets collapsed. Reason: mortgages were given to people who couldn't afford them. This debt was then repackaged and sold to banks and other institutions as good debt. (*The Big Short* by Michael Lewis is an excellent indictment of this time). However, the bigger question remained. Why didn't the financial regulator system catch the problem early, while it was still small?

The answer? **Complexity.**

In *The Dog and the Frisbee* (pdf)¹, Andrew Haldane, Executive Director Financial Stability at the Bank of England, explains all the things a dog would have to know and understand to catch a Frisbee: wind speed and direction, rotational velocity of the Frisbee, atmospheric conditions, and gravitation. It might require a degree in physics to know how to express the control problem involved in catching the Frisbee.

Yet dogs, without physics degrees, do this everyday. They obey a simple rule/heuristic: "run at a speed so that the angle of the gaze to the frisbee remains constant". Empiricism and Simplicity. Agile works because it is an Empirical process using constant feedback to update both the work itself and the way we work.



Image attribution: damedeeso, via photodune

Haldane goes on to show that the financial regulatory system evolved from something simple that many people at a bank could understand, to something only a few people could understand. Eventually it became so complex that no one person understood the system as a whole. The earlier regulatory frameworks worked well in part because many people understood, and therefore many people could spot problems early, before they got too complicated and large to resolve.

As we deal with ever-larger organizations, it's tempting to say that this increase in complexity is okay because we're larger. But if financial crisis taught us anything, the answer should be no. The bigger the system, the more important it is to use simple control mechanisms,

simple feedback loops, and simple measures that can be understood by all. Decreasing complexity — not increasing it — has to be at the heart of all of our decisions. And coupled with that has to be the ability to respond quickly and change appropriately.

Links

1. <http://www.bis.org/review/r120905a.pdf>

Lizzy Morris, CST

Value Adaptations Even When It's "Ouchy"

Original Source: <https://beardeddeagle.com/value-adaptations-even-when-its-ouchy/>

The empirical process of Scrum calls to three pillars: Transparency, Inspection and Adaption. Maintaining each of these pillars takes an effort that comes from the human side of Scrum (i.e., the people, the individuals). Sometimes finding out you need to make adaptations feels like an "ouchy." Many people will say, "Isn't Scrum a team sport?" The answer to that



question is a rousing "yes." Each team is a tapestry of collaboration because every individual brings their skills and depth of experience to the team table.

But what do we do as a team member when we get feedback that places a demand on who we are as a person? What do you do when you have to adapt to ensure the symphony of collabora-

tion doesn't lose its melody?

The challenge of executing Scrum with its heart intact will demand we continuously reflect on ourselves; it demands we allow others to inspect our actions. It means we have allowed ourselves to be vulnerable. Some images pop into our heads with the use of the word vulnerability, and the associated pictures are not necessarily positive.

Let me ask these questions: are adaptations necessary? Are adaptations something that foster the achievement of continued value? The truth is, to be constantly iterative and incremental, we have to become open to adjustments regardless of how much they may stretch us or cause the "ouchy".

I struggle with this question every day because I am in lots of teams: the team of parenting, the team of marriage, the team of coaches — the lists could go on. We are all part of several teams, and they are demanding that we become dynamic and adapt because everyone does not think and process their environment the way we do.

The question left is, will we adapt? The initial adaptation may still be off-key, but the willingness to keep adapting and fine-tuning our collaborative behaviour for our teams will allow us all to experience the music of synergy. There is great value in that adaptation.

Simon Orrell, CEC



Accruing experience with business leaders and Agile environments since 2004, I became one of three Certified Enterprise Coach's in Canada in 2010 and have been working with individuals, teams, and leaders to help transform the world of work. A background including P.Eng. and PMP accreditation also provides me with a pragmatic approach to product, team, and organizational development.

My career in project execution/management since 1996 has spanned a multitude of domains including: manufacturing simulation, digital voting systems, tele-health stations, transportation and warehouse logistics, e-commerce, business process workflow applications, financial transaction systems, and Oil and Gas applications.

I specialize in the initiation and ongoing coaching of Leadership and Agile principles, values and techniques within organizations with an additional emphasis on Product and Portfolio management techniques for both co-located and distributed teams.

I've been a speaker at Agile Alliance, Scrum Alliance, and PMI conferences in both North America and Europe and have recently been invited to speak about applying Scrum values and principles to construction projects at the Agile Summit in Istanbul, Turkey.

Executing Large EPC/EPCM Projects using Scrum Values and Principles

An Experience Paper

What started as an experiment in 2011 has resulted in the successful application of Agile values, principles, and techniques in a domain usually managed with traditional techniques. Prior to starting this work, multiple conversations with the Program sponsor revealed concerns with the team's ability to focus on what's important, resolve issues quickly, and keep commitments. Many of these issues appeared to be rooted in team dynamics and a siloed approach to delivering work product; consequently we decided to experiment with applying Scrum to an Engineering Procurement Construction (EPC) project.

This paper describes how the Scrum framework was applied to this EPC project and 3 subsequent Engineering Procurement Construction Management (EPCM) projects. It is hoped that by describing how Scrum was applied, why it was applied, and the outcomes of that application, the business community at large will see that the Scrum framework is applicable and valuable outside of the software industry. The overarching message of the paper is that the Scrum framework is equipped to help organizations, teams, and people solve common problems — regardless of the domain. The success of organizations, their products, and projects lies in their teams and how they work together.

Introduction

In 2005, a group of Agilists came together to discuss how the Agile Manifesto applied to project management in general. They created the **Declaration of Interdependence**:¹

“Agile and adaptive approaches for linking people, projects and value. We are a community of project leaders that are highly successful at delivering results. To achieve these results:

- We increase return on investment by making continuous flow of value our focus.
- We deliver reliable results by engaging customers in frequent interactions and shared ownership.
- We expect uncertainty and manage for it through iterations, anticipation, and adaptation.
- We unleash creativity and innovation by recognizing that individuals are the ultimate source of value, and creating an environment where they can make a difference.
- We **boost performance** through group accountability for results and shared responsibility for team effectiveness.
- We **improve effectiveness and reliability** through situationally specific strategies, processes and practices.”

This credo was an offshoot of the **Agile Manifesto**² created specifically for software development in 2001 by some of the same people. Since 2004, I've been working with teams in different domains within the software industry applying Agility (and specifically Scrum) to the execution of projects and product development.

In late 2010, a large North American energy company embarked on a large Engineering,

Procurement and Construction (EPC) project to build a natural gas processing plant in northern British Columbia, Canada. The plant was to have the capacity to process 400 MCF/day and the plant was to be on-line in December 2012. The budget for the project was \$700M. In April 2011, the project manager and I started having conversations about his team's struggle to move certain activities forward, and the specific problems his team was facing:

- Unclear priorities and roles
- Lack of communication due to silo'd activities
- Lack of visible progress on what's important and why
- Lack of accountability to each other
- Lack of understanding of the true state of the project

These problems are common with many teams I've worked with in other domains. These are typical problems that can be addressed based on the 3 pillars of a Scrum mindset.

- Transparency
- Inspection
- Adaptation

We decided to experiment with applying Scrum values, principles and techniques to address these problems on the EPC project.

Change in the Context of Physical Construction

Going into this work, I had assumed that change was difficult in these types of projects because of the physical nature of the work. Once equipment or modules are fabricated, it is difficult to change them. This turns out not to be precisely the case. The construction activities of an EPC project are extremely well understood. Relative to the engineering and procurement activities, there is significantly less risk in the construction activities. Change is difficult due to all the review/approval processes necessary to finalize design, engineering, and procurement. When there are delays in reviews and approvals regarding engineering or procurement, the effect is amplified:

- Materials and equipment don't get purchased fast enough
- Which leads to those materials and equipment not arriving on time
- Which leads to less time for construction
- Which leads to missing the online date.
- Which leads to decreased revenue and therefore decreased ROI

The revenue and ROI model of a natural gas processing plant is much better understood than the analog models for software products. Therefore, it is much easier to understand the true cost of delay. The cost of delay for the initial EPC project was \$300k/day. The overall challenge within these projects is to minimize delays between dependencies to allow maximum time to deal with "the unknowns you don't know" as well as the "unknowns you know".

Application: Engineering, Procurement, and Construction Project

For the initial EPC project, the energy company had contracted an engineering company to provide the design and engineering, contracted a Works contractor to construct the plant, and self managed the project including the procurement. The approach to solving the initial problems exhibited by the team was to base teamwork on the **Scrum values: Focus,**

Courage, Openness, Respect and Commitment.

The team needed to find a way to **focus** on what issues were the most important for the success of the project. We achieved this by implementing a backlog of issues that were iteratively prioritized by the project manager. The combination of making project prioritization visible (rather than discipline prioritization) and limiting Work in Progress (WIP), the team was able to continually focus on what was important for the project's success.

The team needed to have the **courage** to face both the reality of their progress and the barriers to their effectiveness as a team. We increased this courage by implementing iterative feedback loops for both the work and the process used to achieve the work product. The retrospective sessions focused on how the team was working together. This resulted in the team taking ownership of several actions per week to improve their communication, collaboration, and work processes. Iterative planning sessions were held based on progress achieved (as demonstrated in bi-weekly schedule reviews) and roadblocks anticipated in upcoming work.

The team needed to be **open** to experimenting with different approaches to their work. They also needed to be open to the transparency that using Scrum values would create. All of the dysfunction in the team would be surfaced and would need to be addressed by the team.

The team needed to **respect** and trust each other in order to perform effectively. To this end they created basic working agreements and work processes which relied on trust.

The team needed to make and keep their **commitments** to each other. In the face of their individual discipline priorities, they had to commit to prioritizing the project's commitments in the context of collaborative multi-discipline. Essentially, they needed to regularly commit to optimizing for the project, not their individual disciplines. Then they needed to see those commitments through.

In the context of the 3 Pillars of Scrum, the values described above were promoted through the application of Scrum Roles, Events, and Artifacts as described in the latest **Scrum Guide**.³

Scrum Roles

Product Owner

An existing role in the energy company's organization called the **Project Manager** was responsible for regularly establishing the priorities and context for the team. That person had the overall accountability for the success of the project and was already expected to have an overall understanding of the state of the project and the progress and roadblocks to success.

Scrum Master

We created a new role we called a **Team Facilitator** who was responsible for helping the team keep an Agile mindset and coordinate the transparency of goals, progress and impediments. This person facilitated the daily standups, the iterative planning meetings, and the retrospective sessions.

Team

The existing team consisted of a group of multidisciplinary specialists (Engineering, Procurement, Project Controls, Documentation, HSE, Operations, QA, Contracts) respon-

sible for producing valuable deliverables to be consumed by the Procurement and eventually the Constructors. They were organized in a matrixed fashion, reporting to discipline leads in terms of how they performed their work but reporting to the Project Manager in terms of what they worked on.

Scrum Events

Sprint

As illustrated in Figure 1 (*see Appendix*), traditionally managed construction projects rely on monthly project schedule updates to inform a 12-week forecast, which in turn informs a 3 week forecast which sets the high level activities for the immediate 3 weeks. Progress on those activities are then reviewed once per week. We implemented a 4 week **iteration** which took priorities from the 3 week look ahead and other sources, created task plans for the priorities, and then monitored progress and roadblocks daily. Team members were instructed to keep WIP to a minimum by always working on the highest priority tasks they had taken on.

Sprint Planning

Every 4 weeks the team participated in an **iteration planning** session. Prior to the planning session, priorities were established at a backlog refinement session by the Project Manager who considered elements from the schedule, infrastructure requirements, and retrospective actions. At the planning session, an owner for each priority was established and a task plan (which involved the participation from multiple disciplines, complete with task owners and duration estimates) was created. That plan was then inspected and adapted on a daily basis as new information came to light. We did not focus on the concept of ‘velocity’ at all. Often teams need to be able to assess how much work they can accomplish in a given iteration. In our case, we knew what we had to accomplish; we needed to be able to identify and address roadblocks and risks as expeditiously as possible in order to meet our deliverable goals.

Daily Scrum

On a daily basis, the team met in a **standup** format for 15 minutes to discuss the progress and roadblocks to the iteration priorities. The team met in a ‘situation room’ where the iteration goal, priorities, plans, progress, and roadblocks were visualized on the wall using index cards and poster paper. Figures 2 through 6 (*see Appendix*) illustrate examples of those artifacts. The team members discussed progress of tasks and roadblocks to progress. The team inspected a task hour burndown as another signal of roadblocks. Those roadblocks were noted and assigned to the Team facilitator for removal.

Sprint Review

The team participated in a bi-weekly schedule review of what had been actualized in the project schedule. This review gave the team and management the opportunity to review completed deliverables and informed the need for new items to be included in the backlog.

Sprint Retrospective

At the end of every iteration, the team participated in a retrospective session to discuss what had gone well and what could be improved. The team used the results of their previous iteration to speak about what they should keep doing, what they should start doing and what they should stop doing. It also provided a forum for the team to discuss how they were communicating and collaborating and what they wanted to change about their working agreements.

Artifacts

Product Backlog

A backlog of items called the **Project Focus Backlog** was created from multiple sources of issues. Level 3 Schedule progress, roadblocks to meeting deliverables in the schedule, risks from the risk log, process infrastructure, and retrospective actions were all used as sources for backlog items. To identify new backlog items, the team used a mind mapping technique to answer questions like “What is preventing us from understanding the true state of the project?” and “What is preventing us from understanding the project Estimate At Completion (EAC)?” Answers to these questions allowed the team to assess root causes of existing issues and the priority for solving them.

Sprint Backlog

Every three weeks the highest priority issues (as determined by the Project Manager) from the Project Focus Backlog were added to the **Iteration backlog**. That iteration backlog was then used as the focus for the team to create task plans for the completion of the issue. Further the iteration backlog was one of the central tools that the team inspected at the daily standup.

Product Increment

Perhaps the most difficult aspect of Scrum is the creation of a product increment every sprint. A product is simply a collection of solutions to business problems for which someone is prepared to pay. In software, the product increment is usually defined as working, tested functionality that provides value but may not be shipped to the customer yet because not enough value has been accrued. On an EPC project, because Construction is the least risky part of the endeavour, the product increment took several forms: a 3D model of the plant and the deliverables necessary to construct the plant. The 3D model of the plant was reviewed iteratively based on the latest design documents and engineering. Constructability analysis of the model then led to changes in design or procurement. The deliverables necessary to construct the plant were the Engineering Drawings and Procurement contracts and purchase orders necessary to get the materials and people to site. With these deliverables in place according to schedule, construction activities could proceed to meet the online date. In a pure sense, the analog to the integration testing necessary in software development to provide working, tested software is the finished plant construction.

During the ‘Turnover’ phase of construction a multi-discipline team iteratively and incrementally ‘turned the plant over’ by testing and verifying one subsystem at a time. Turnover, final testing, and commissioning constituted the ‘working, tested’ product increment. Once all sub-systems had been turned over, the plant was complete and ready for operation.

Artifact Transparency

As previously mentioned, the team held all their daily standups, planning sessions, and retrospectives in a “Situation Room”. The project and sprint goals, along with reminders of working and communication agreements were on the walls. The iteration backlog took the form of a Kanban board composed of issues and their respective tasks on the wall. Numerous mindmaps were also up on the wall so that the team could also see their progress in meeting their goals outside the context of the project schedule. Figures 2 through 7 (*see Appendix*) illustrate the use of the situation room. When it was necessary to have people attend meetings remotely, they would do so via conference telephone while looking at photos

of the kanban board.

Results of Applying Scrum Values to the Project

The benefits of executing the project using Scrum values were many:

- We were able to ensure that the team regularly understood what the project priorities were that superseded their respective discipline priorities. This was evidenced by participants being able to speak about conflicts with their discipline priorities.
- We were able to gauge progress based on empirical evidence; what valuable deliverables had been completed. By developing task plans for each project priority, we were able to gauge our progress on a daily basis rather than believing "it's on track" until the day it is late.
- Challenges, roadblocks, and issues were exposed quickly (daily) rather than remaining hidden for weeks at a time. The quicker exposure allowed more time to address each of them. Priority issue cycle time was reduced by a factor of 3.
- By encouraging the team to limit WIP, we were able to continually complete items in the backlog of issues rather than having many items in progress but none complete.
- Team composition shifted early on as some team members were uncomfortable with the openness and accountability required. As a result, we had a group of people who shared similar values and could come to agreement on how to work together. Those shared values enabled the team to iteratively improve on their communication, collaboration and work processes.
- Overall Project Cycle time reduced by 5-10% as estimated by the Project Manager.
- The project management team was able to maintain the original project schedule, complete construction on time, and stay within the Class 3 estimate. This in the face of projects of this type and over \$500M being on average 48% over budget and 18% late according to 2010 Independent Project Analysis data (see Figure 8 in the Appendix, page 116).
- The plant was sold before it went on-line.

Application: Engineering, Procurement, and Construction Management Program of Projects

After the success of the application of Scrum values to the first project, the energy company selected a similar approach to executing a Program of three large EPCM gas plant construction projects in Northern BC. An EPCM project is one in which an EPC Contractor provides management services for all aspects of the project from design and engineering through to construction and turnover of the plant to the owner. In this scenario, the energy company is ostensibly simply in an oversight role. The Class 2 estimate for the program was ~\$3B.

In this Program there were to be several unusual aspects:

- The concept of "design one- build many" was to be employed. This meant that even though each of the plants would have different design conditions, designing components to a suitable base case and having many of those components shared amongst the plants would be economical.
- The schedule would be compressed such that there was significant overlap between projects in their respective Engineering, Procurement and Construction phases. Just enough engineering would be completed to allow procurement to start and similarly just enough engineering and procurement to allow construction to commence. This principle was being applied across three projects with EPC phases staggered only enough to accommodate the engineering and procurement design reuse.

- Finally, one of the plants would be an extension to an existing plant which meant that construction activities would be considered 'brown field' and occur around an already operating plant.

The major difference of this program from the initial self-managed project was that multiple / competing cultures were in play. While the initial EPC project had the owner, an engineering company, and a construction contractor involved, they were all managed by the owner. For the Program of EPCM projects, the EPC contractor managed the relationship with the construction contractor (and all other contractors) and the owner was simply in an oversight role. This proved to be the most difficult aspect of the program to manage.

Initially our approach was to use Scrum to manage oversight activities. In short order it became clear that was not going to provide the visibility the team needed. The issue was that the EPC contractor did not share many of the owner's values and didn't want to adopt any of their methods. In retrospect the Owner and EPC contractor cultures were quite different as illustrated by the Competing Values Framework analysis in Figure 9 (*see Appendix*). Some examples of how the cultures differed could be seen in symptoms like:

- One organization did not want posters on the walls and was reluctant to publish true status (or potentially any negative connotation) on LCD TVs, while another was trying to make messages as visible as possible.
- One organization tried to be very mindful of how calendaring was used to coordinate meetings, while the other had an "accept everything and decide last minute" approach to meetings.
- One organization was fearful of open disagreement, while the other organization thought it was necessary.

Rather than shift the cultures one way or the other, we needed to help create a culture for the program that both parties wanted to live and respect. The owner had engaged me as a member of a small group of 3 specialists in Communications and Leadership coaching. This support group decided to utilize an approach from Patrick Lencioni's *The Advantage* to start that process of growing a new culture.

This support group focused on helping the owner and EPC management jointly create a guiding framework based on a Mission, Vision and Values for the Program. The process of creating that guiding framework led to an agreement to working as a singular team, which included joint participation in the Scrum activities. Further, a Scrum of Scrums approach was taken for the Program. The owner, EPC Project managers, and functional managers held their own Scrum activities based on a Program backlog containing program-wide priorities and roadblocks. Both the Project and Program activities were initially facilitated by the owner.

While the owner was substantially co-located inside the EPC contractors' offices, the overlapping nature of the projects and the overlapping phases of each project led to the need to apply techniques for distributed Scrum activities. A situation room was no longer feasible (for cultural reasons) and the artifacts for Scrum activities needed to be visible remotely (from other offices, the Module Yard, and site) using screen sharing applications during standups, planning and retrospectives.

Breaking Through – “Necessity is the mother of invention”

In December 2015, the owner and EPC contractor leadership gathered to discuss the current state of the Program and its future. The Leadership agreed that the projects making

up the Program were very likely going to be 10% over budget and 2-3 months late if the Program continued to function as it had over the past year. These predictions were in-line with average performance by the industry in western Canada. At that time, Leadership made a commitment to each other to perform well above this industry average; to be part of something extraordinary. This level of performance would both solidify the relationship between the EPC contractor and the owner, but also likely bring further investment to the Program from 3rd parties attracted by that performance. To measure this performance, Leadership set themselves the following targets for the Program:

- Meet Plant design standards;
- Safety Total Recordable Incident Frequency (TRIF) < 0.1;
- Plant A – 1 month early & 10% lower Total Installed Cost (TIC);
- Plant B – 1.5 months early and 15% lower TIC;
- Plant C – 2.5 months early and 15% lower TIC.

In order to achieve these targets, the leadership team recognized that the way the entire Program worked towards goals was going to have to change. They committed to the following principles which are the foundation of supporting extraordinary performance:

- We challenge the status quo;
- We focus on outcomes;
- We make commitments mindfully and honor them;
- We consider these 4 key factors first:
 1. safety
 2. schedule
 3. quality
 4. cost

Average performance is characterized by being predictable based on what we know. Innovation and extraordinary performance arises when we allow ourselves to commit to outcomes that aren't part of what we think of as predictable. We don't know how we're going to achieve the outcome, but we believe the outcome is possible, and we are committing to finding out how it is possible and executing on that discovery.

“If you solve one problem, and then the next one ... if you solve enough problems, you get to come home.”

- The Martian

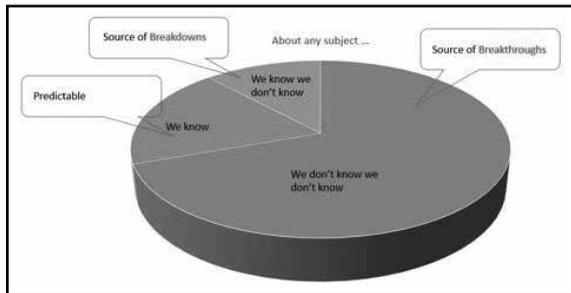
Achieving extraordinary performance involves the following steps:

- Set a vision
- Enlist in the vision
- Identify breakdowns
- Manage the breakdowns to create breakthroughs

The next few sections of this paper describe the process by which we helped the team members reach extraordinary performance.

For the Program, the overall vision had been set by Leadership. They had committed to enabling the Program to meet the extraordinary goals. To do so, they had also committed to focusing on true “leadership” rather than “management” — where “management” is defined

as minimizing risk and maximizing predictability, and “leadership” is defined as enabling the creation of something currently unpredictable. It is the role of Leadership to continually enlist recommitment in the vision. Leaders speak in declarations of what is possible and what will be. This is fundamentally different from the language of Management which is the assertion of what is predictable based on historical evidence.



“Breakdowns” are defined as any gap that exists between what has been committed to and what is currently predictable. Breakdowns should be embraced as opportunities for innovation and extraordinary performance. “De-committing” is the act of allowing a set of circumstances to distract the team from honouring their commitment. A

“Breakthrough” is the result of managing a Breakdown (avoiding de-commitment) so that the gap between what is predictable and what is committed to is eliminated.

Enlisting Team Members in the Vision

While the Program Leadership had voluntarily committed to meeting their extraordinary performance targets, Leadership needed to continue to garner voluntary and personal commitment to those goals from all team members. Voluntary personal commitment is only possible if team members are free NOT to commit. Assessing levels of commitment and garnering personal commitment involves the continuous application of the most powerful medium of leadership; conversation. Leaders must take every opportunity to engage in conversation (most effectively face-to-face) with their team members in order to provide them with the guidance, safety, and a model for making and meeting commitments. Those who have not yet committed can prove useful during the management of breakdowns, as they can provide necessary balance during conversations. In the interim, it may be enough for Team members to simply believe that the Program goals are possible. Helping the team come up with their specific version of the vision will aid greatly in garnering commitment from the team.

Identifying Breakdowns within the Program

In order to execute on this model effectively, Breakdowns needed to be identified and managed iteratively. While the horizon for the commitment of the program was 18 months in the future, the ability to meet those commitments was contingent on keeping commitments with nearer time horizons. Inspecting for Breakdowns (and managing them) early and often was the key to successfully achieving extraordinary performance.

For the Program, that iterative Breakdown identification process involved decomposing longer term milestones down into nearer term outcomes. Goal X could only be achieved in 18 months if interim goals A and B were completed in 6 months, and goal A could only be achieved if goal K was achieved in 1 month. Breakdowns could have occurred at any or all of these horizons.

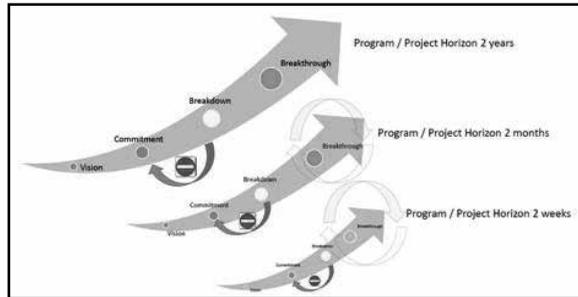
Project managers, functional managers and team members needed to inspect the L2, L3 and L4 schedules at least monthly to identify those schedule-based outcomes where there was uncertainty as to how they will be achieved. Any team member could declare a Break-

down. These Breakdowns were compiled and prioritized by their respective Project Manager. Breakdowns deemed to be applicable to multiple projects or unworkable at the Project level were identified as Program Breakdowns.

Creating an Environment to Surface Breakdowns

The base mechanics of identifying Breakdowns were already in place within the Program. Nominally the agile framework of Scrum was in place. This framework involved each Project team iteratively identifying their priorities, planning the work necessary to commit to achieving those priorities, inspecting and adapting those plans daily based on progress and roadblocks, and then reviewing their work process/culture for improvements before repeating the cycle.

Nothing about this framework needed to change except the rigour by which outcomes were identified, prioritized and planned. If teams had not been using 14 day, 30 day, and 60 day look aheads as well as L2, L3 (and eventually L4) schedules as inputs into their priority identification, they were now required to do so. Project Managers were expected to manage the ‘backlog’ of these priorities (now containing Breakdowns) and teams needed to spend more time planning their upcoming work activities to meet the outcomes. Teams decided for themselves how much time to spend in planning, but 5-10% was not unreasonable. Some teams found this iterative process more amenable every 2 weeks, while others preferred every 4 weeks.



However the Breakdowns were identified, their widespread dissemination, communication, and visibility was crucial to ensure that all people who might have ideas on how to challenge the status quo were aware of the Breakdown.

However the Breakdowns were identified, their widespread dissemination, communication, and visibility was crucial to ensure that all people who might have ideas on how to challenge the status quo were aware of the Breakdown.

Creating an Environment to Manage Breakdowns

Focus, Courage, Openness, Commitment, and Respect

A Breakdown is a gap between what has been committed to and what is predictable. Once a Breakdown has been identified and prioritized as being significant enough to work immediately, a person must be identified as the Breakdown leader. That person took ownership of the coordination of the planning and activities that needed to occur for the breakdown to be managed to a Breakthrough. Usually this involved coordinating multiple discussions with multiple team members who have experience with the intricacies of the breakdown gap. These discussions focused on challenging the status quo, but took many forms/combinations including but not limited to:

- Business process workflow value analysis;
- The 5 Whys;
- Socratic Method.

These conversations were time-consuming, and that time was prioritized according to the outcomes' importance for the Project or Program to meet their extraordinary targets.

The success of these Breakdown conversations relied primarily on three factors:

- having the right people involved in the conversation;
- having enough time to resolve the breakdown before it materially affects the Project/Program;
- the perseverance of the participants to solve the Breakdown.

Conversations were limited to 6 people and a facilitator. We made it clear that titles were not an indicator of the suitability to work a particular Breakdown. While it was sometimes useful to include people directly involved in the work process that was untenable, it was also useful to invite those who were completely unfamiliar with the process. These people were able to ask questions that others closer to the matter sometimes weren't capable of seeing in the moment.

The faster a Breakdown was identified and prioritized high enough to work immediately, the longer the team had to transform the Breakdown into a Breakthrough. There is an inverse relationship between the frequency with which Breakdowns are identified and the time available to work the Breakdown. For example, if Breakdowns are only identified monthly, it is possible that an entire month will be lost to the effort of solving that Breakdown. The use of the "24-hr rule" helped maximize the time available to think about and solve Breakdowns.

There was no guarantee that a Breakdown would lead to a Breakthrough. Honoring a commitment mindfully simply means that either the commitment is met, or that the team is notified as quickly as possible that the commitment will not be met. This notification allowed those who are depending on it to adjust as best as possible. Judgement was required to assess diminishing returns on efforts spent looking for a resolution, but we wanted the team to exhaust challenges to status quo before considering de-commitment.

Avoiding De-Commitment

The most difficult part of managing Breakdowns is avoiding de-commitment. There are many organizational, cultural, and personal mechanisms that we allow ourselves to be impeded by. Part of challenging the status quo involved challenging those mechanisms which had become normal work and communication practices. Some examples included:

- treating the sending of an email as a transfer of ownership;
- allowing the apparently urgent to distract from the important;
- notifying people of a missed commitment the day the commitment was due;
- communicating apparent commitments without due diligence;
- responding to challenges to status quo with "That's not how we work".

In any team member's toolbox, maybe the most important tool to deal with de-commitment is courage — the courage to speak out when seeing other team members de-committing. Part of modelling leadership is the ability to speak with honesty and respect about our propensity to de-commit. Modelling this behaviour at the highest levels encourages and enables others to adopt this behaviour.

Escalating Breakdowns

Escalation of Breakdowns occurred as a normal part of the Breakdown management process. As potential solutions involving new ways of working were surfaced, it was normal for team members (and specifically the Breakdown lead) to want to gain "permission" from their management and/or Leadership. It was important for Leadership to sanction these potential

solutions as soon as practical so that the team could move forward with their solution toward the realization of their Breakthrough.

During the identification and prioritization process, Project Managers escalated Breakdowns to Program Leadership when they saw that the Breakdown impacted multiple projects. This escalation required the coordination of the Project Managers and the Functional Managers. Iteratively, Program Leadership prioritized the Program Breakdowns, assigned Breakdown leaders, and managed the Breakdowns toward Breakthroughs.

In order to help carry momentum of this process forward, we encouraged team members to widely communicate and disseminate the results of extraordinary efforts to achieve Breakthroughs. Doing this as quickly as possible (within 1 day of a Breakthrough) helped others who were struggling to avoid de-commitment to see what was possible.

Results of Applying Scrum Values to the Program

- Owner's Program Director has stated that the overall Program cycle time has been compressed by 10-15% compared to original schedule resulting in savings of between \$200 million and \$300 million.
- Currently on pace to deliver ahead of schedule and under budget in a domain where 30% over Class 3 budget and schedule is the norm.

From a team survey:

- "When there are challenges, the team is encumbered by processes and procedures and linear thinking. The result of applying Agile and breakdowns is the realization that challenging the status quo and using non-linear thinking is required for success. This doesn't show up in a Gantt chart."
- "I used to come to work facing problems that looked insurmountable, now we are on the precipice of something extraordinary and the problems look solvable."

Owner's Deputy Program Director stated in his Lessons Learned about the Program:

- Only work with an EPCM contractor who uses an Agile approach;
- Provide support for Agile and breakthrough approach.

Lessons Learned

- Shifting from using Scrum for the Owner's oversight activities to using Scrum for the joint/execution should have happened sooner. This delay occurred because it took so long to create a trusting relationship and culture, which ultimately was about placing the right people in the right roles. If that had taken place, the transition of ownership of the Scrum approach to the EPCM contractor would have occurred sooner.
- Eventually, it became clear that the ownership of the Scrum activities needed to be assumed by the EPC contractor to be truly effective. This transition relied heavily on the personality and training of the team facilitator from the EPC contractor. We knew the transition had occurred successfully when the Scrum activities were occurring without the requirement for Owner instigation and changes were being made based on the needs of the team.
- The final transition of the Scrum activities occurred when the centre of gravity shifted from the EPC contractor offices to site. Once again, a key part of this transition was finding the right people to champion the Scrum activities and the training of all involved.
- The inability to make use of a central "situation room" degraded the communication at Scrum activities. The use of Excel, Webex, Skype, and telephone allowed for the distributed participation during those activities, but the interactions were not as rich as in a room.

- Task duration estimates were deemed not relevant in this context. What was more important was the regular visibility of what needed to be done and why to help with prioritization and capacity discussions.
- The Scrum of Scrums approach to the Program standup effectively exposed Program-wide issues and concerns that affected multiple projects. It also exposed when it was taking too long to deal with those issues. Finally, it promoted the transfer of knowledge from one project to the next as the approach progressed through similar phases of execution.
- Use of the "24-hr rule" was critical to exposing issues and roadblocks as quickly as possible and led to increased time for the team to solve them.
- Applying the concept of "Just Enough" to Model reviews was effective. Because engineering overlapped with Procurement which overlapped with Construction, model reviews which are usually done once a certain percentage of engineering is complete, were instead performed once enough information was available to make Procurement and Construction decisions. Model reviews became more iterative, smaller, and more frequent. Fewer drawings were required in order to meet the goal of the reviews — just enough data for the purpose of the Model Review. For instance, in order to facilitate the early works activities, only weights and placements for piling were necessary to be part of a model review. This allowed procurement to order earlier than normal, and thus construction could start earlier than normal.

Despite the existence of a detailed schedule, it was not used the way we were led to believe. It suffers exactly the same flaws as trying to use a schedule in software development, with the exception that it outlines a greater percentage of the activities that need to be completed for success. It is created by people not familiar enough with the execution of the work to vet the logic and estimations. Because it is so complex (4000 lines), it is unwieldy to update regularly with the latest information.

The Future of Agility in EPC/M Projects

The Agile Manifesto, originally created in the context of software development, is easily modified to apply to any domain. Similarly, the 12 Principles behind the manifesto can be easily adapted and applied outside of the software industry. Finally, the goals and approach put forth by the Declaration of Interdependence has proven applicable in the field of EPC/M projects. Our experience with these large natural gas processing plant construction projects has corroborated these assertions.

To fully realize the value of using Agile frameworks like Scrum, the industry needs to shift from a static mindset to one that values continual learning and challenging the status quo. While implementing the mechanics of Scrum provides increases in effectiveness and efficiency, it is an Agile *mindset* adopted by more and more members of the teams that will lead to more significant benefits.

In the application of Agile values and principles in software development, one of the core tenets is to use working/tested functionality as the true measure of progress. With this measure of progress, the business is able to routinely make functionality vs schedule decisions.

While building gas plants, EPCM organizations use completed engineering and procurement deliverables as the measure of progress. Comparing that actual progress against the estimated progress articulated in a project schedule provides the business with an indication of how they are performing against their time and budget goals. In these projects, functionality is not the main lever of course correction; schedule and cost are the main levers.

In Agile software teams, emphasis is placed on always knowing the current state of the software; what valuable functionality could we ship to customers if necessary. To know this, Agile teams use continuous integration, automated builds, and automated testing to minimize the time it takes to understand the state of the software. Agile teams often start with a substantial amount of manual testing which can take days or weeks to execute in order to understand the software state, however, as Agile teams mature they strive to have a much more frequent (daily) view of the software state through the implementation of automation. This more frequent view is also used to uncover problems (as well as progress) which can then be prioritized and addressed in the context of value. At the very least, truly Agile software teams KNOW the true current state of the software every 2 weeks.

In order for an EPCM organization to have a true state of progress, they would have to consider changes such as:

- Automation of the deliverable progress update process to minimize the time it takes to understand progress against the schedule. Currently many manual updates are made which provide a progress view every 2 weeks but with data at least 1 week old. This improvement would be akin to the continuous integration/testing required on an agile software team; knowing as often as possible the true state of progress.
- Iteratively reviewing and updating the schedule logic to ensure it reflects the latest reality. In combination with using a floating and fluctuating project end date (which makes effects of changes visible immediately) this would be akin to the iterative release planning an agile software team performs.
- Using this information to rigorously and continuously look ahead to identify and innovate around the issues that need resolving in order to maintain the construction schedule.
- Experimentation with measuring the cycle time from Engineering through Construction for discrete activities starting with Early Works activities (civil, roadworks, fencing, piling). By looking at that cycle time, it may become easier to expose and address unnecessary additions to that time which can then be measured on future activities.

Links

1. <https://pmdoi.org/>
2. <http://agilemanifesto.org/>
3. <http://www.scrumguides.org/scrum-guide.html>

Appendix

Figure 1: Agile Construction diagram

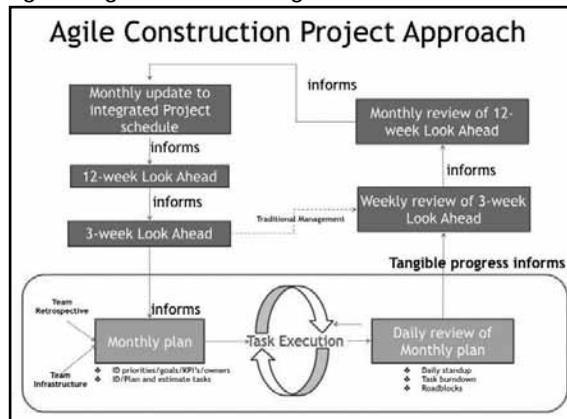


Figure 2: Communication Agreements

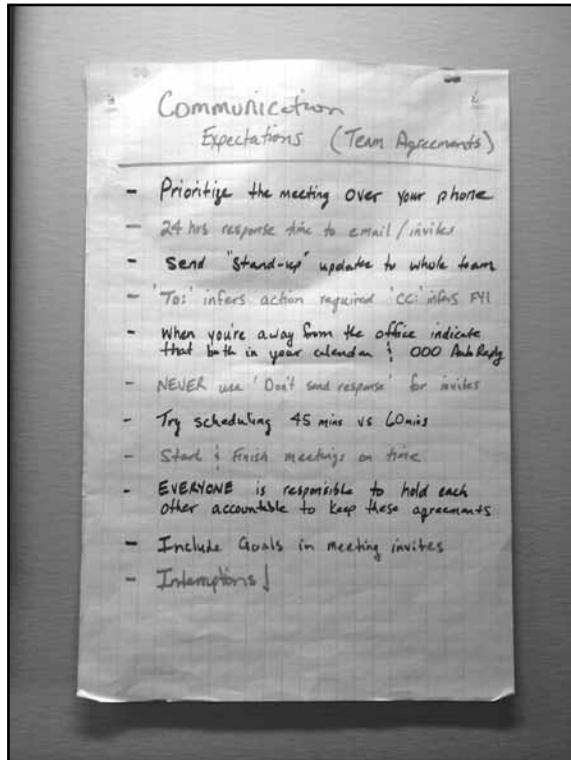


Figure 3: Retrospective Actions

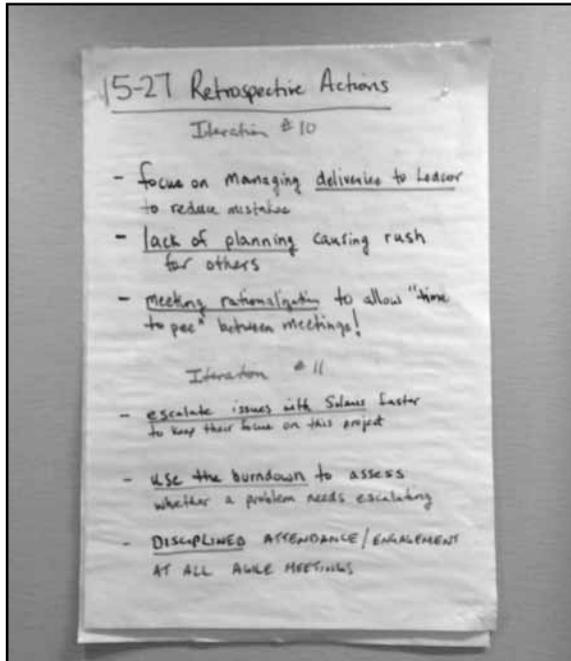


Figure 4: Kanban and Burndown

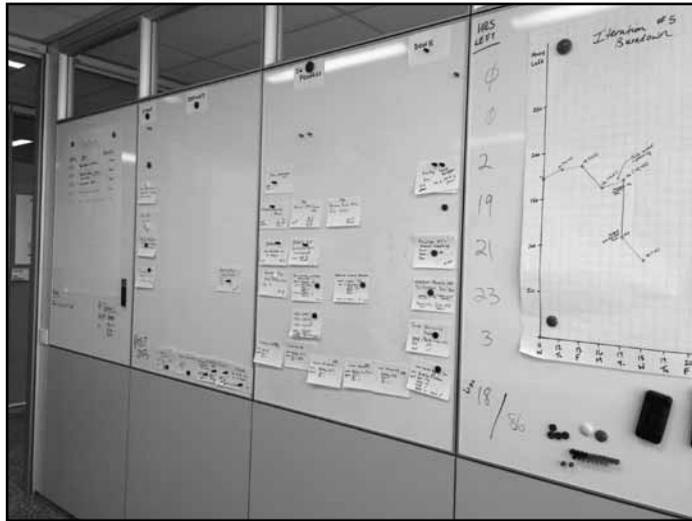


Figure 5: impediments

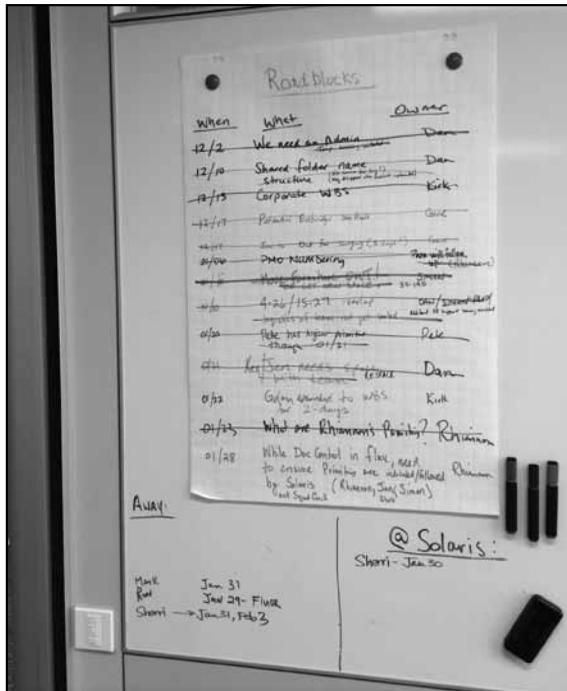


Figure 6: Situation Room



Figure 7: Mind Mapping

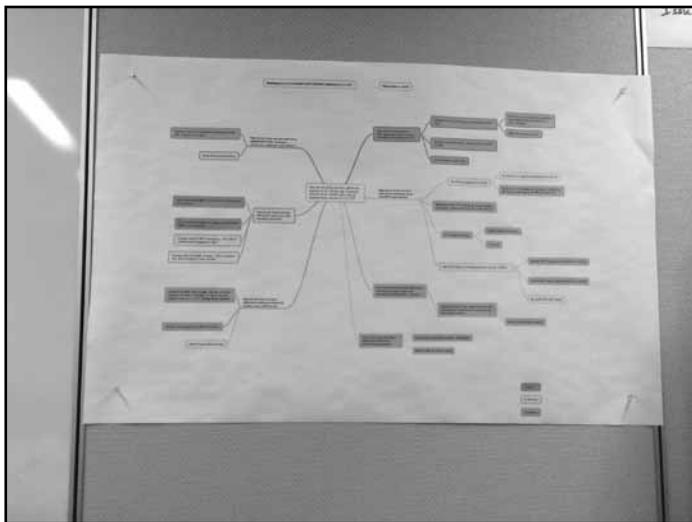


Figure 8: Independent Project Analysis chart

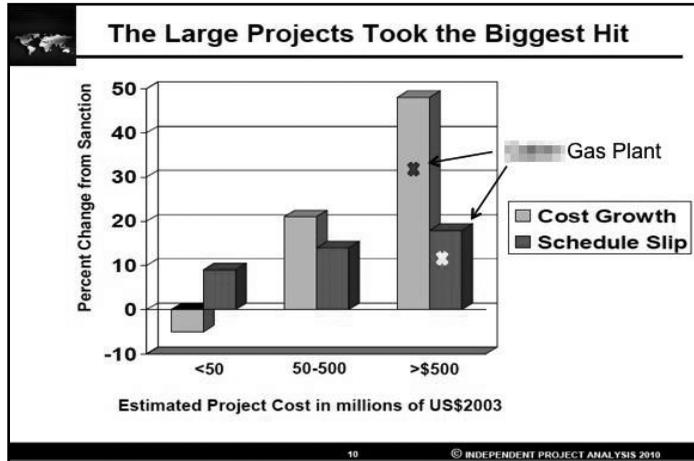
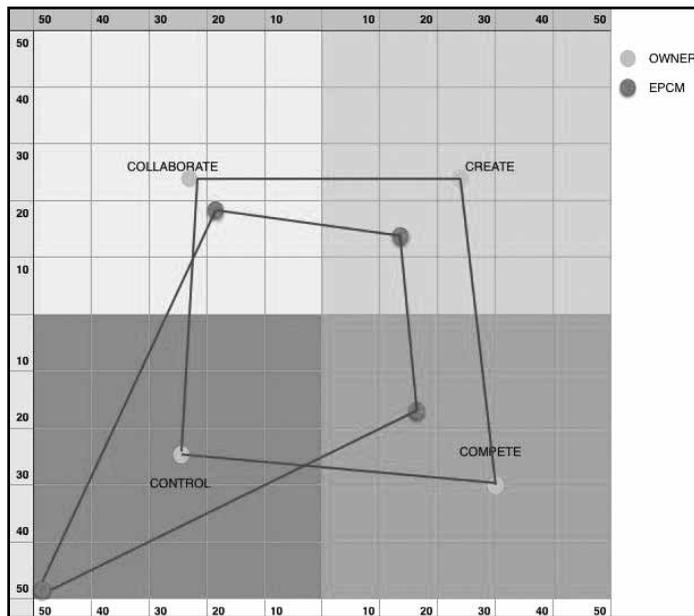


Figure 9: Competing Values Framework Analysis



Dhaval Panchal, CEC, CST



Dhaval Panchal is an experienced Executive and Agile Coach. He is a ScrumAlliance Certified Enterprise Coach (CEC) and a Certified Scrum Trainer (CST) with 15+ years of Agile experience working in the development and management of products and services in the software industry. He has experience working with startups to Fortune 100 companies, with clients in the telecommunications, business process engineering, shipping, e-discovery, legal, gaming, health insurance, and oil and gas industries.

As an Agile coach, Dhaval brings a results oriented, people-centric perspective. He is often described as a creative simplifier who is able to navigate clients through a variety of challenging contexts. Dhaval helps his clients to focus on signals-over-noise and exposes trade-offs that client organizations have to make in the “real world”. Dhaval has presented at various Agile industry conferences and is sought after by leadership teams to make sense of complex organizational situations.

Dhaval Panchal delivers world class Agile Coaching and training services. He works with CTOs and heads of software development, engineering, DevOps, data & project management to:

- Transform legacy waterfall approaches into effective Agile processes;
- Scale Agile efforts to full enterprise level;
- Train key team members to maximize Agile adoption;
- Adopt practices to ensure sustainable momentum and long term success.

He can be reached via email at dhaval.r.panchal@gmail.com and on Twitter @dhavalpanchal.

My Favorite Coaching Experience

Original Source: <http://www.dhavalpanchal.com/my-favorite-coaching-experience/>

In 2009, my good friend Subhayu was visiting me in Seattle. In India we would often hike together through remote hills in Western Ghats. So it seemed appropriate that I sign up myself, Subhayu and another friend of mine from Seattle, Ben, for a day of adventure. Whitewater rafting seemed appealing at that moment. My friends had no prior experience with whitewater rafting. My adventurous self had only once been through the Skykomish river **Class IV and Class V rapids**¹, wherein my group avoided some of the dreaded Class V rapids and walked our rafts along the shore. This time, however, I wanted the real deal and signed with a professional guide who would coach and guide us through Class IV and Class V rapids on the White Salmon River. To my excitement, White Salmon River rafters have an option to paddle through air while falling fourteen feet over Husum Falls.



Husum Falls, WA

Subhayu, Ben and I reached the rendezvous point on time. We parked our car, checked into wet suits, and signed release forms. We drove in a shuttle to the launch site where we were provided a very important safety talk by our guides — which, and I blame it on too many air-

plane flights, I did not pay much attention to. The agency I had signed us up with had many professional guides and many other people like us, so we were divided into groups of six with one guide per raft. My group consisted of my friends and three other guys whom I had never met before.

The first hour on our trip was as gentle as whitewater rafting can be. During this period, our guide patiently explained how we should position ourselves on the raft and how to paddle through water. She explained some voice commands and we practiced steering our raft as per her guidance. Initially, we struggled a lot with our raft practically going nowhere. However, as we practiced and practiced, our group got the hang of it.

The next hour and a half was far more challenging, full of excitement with twists and turns often spinning our raft 360 degrees. We soon realized that, unlike typical boats, rafts in

choppy whitewater do not have a fixed bow and stern. The situation changes all the time, where in one moment you are in the front and the next moment you are at the back end of the raft. Over the crash of the waves, screams, big boulders and near misses we stayed tuned to voice commands from our guide. She did a great job of keeping her head above the fear and thrill of the moment to harness our energy towards an exciting ride (thus far). During brief moments of lull, she would tell us stories of other trips she had taken through these waters. These ranged from pleasant stories of wildlife sightings to terrifying rescues of overturned rafts. We had our first scare while navigating around a big boulder. Subhayu lost balance and was hanging upside down with only one of his feet in the raft and the rest of

And she said, “you don’t want that” – in a tone reflecting her motherly meanness.

him getting tossed around in the water. Through combined effort from a couple of group members, we were able to pull him back up into the boat – disaster averted! We played with a few minor scares wherein later Subhayu grabbed me just in time to save me from the experience of a chilled water head-first dive.

In retrospect, these scares prepared us well for what Husum Falls had in store. Husum Falls, with its fourteen feet drop, is the cherry on top this cake. This is why I had dragged my friends along. Prior to negotiating the falls, we rested our raft on the shore, walked a bit to visually inspect the falls and pep talk each other to sign up for the adventure. Having secured our agreement, our guide coached us for the specifics of rafting over this insane drop. We were to paddle until we caught the current, then steer to get the right angle of approach for the falls. Then, when she yelled, we all were to crouch down with our paddles rested to cling as close to the floor of the raft as possible. This last bit about crouching was very important because as the raft hits the bottom of the falls, it behaves as a compressed spring. First bending and then springing open to regain its shape, this rubber band effect is strong enough to flip people overboard.

And she said, “you don’t want that” – in a tone reflecting her motherly meanness.

So, we earnestly practiced a couple of dry runs to get the crouching part right. She observed and corrected us. We were now ready and just in time, since the current was now pulling us rapidly.



Throw Bag - Safety Rope

Our guide steered our raft, just as she said she would. She positioned us for the angle of approach, just as she said she would. We crouched to the bottom of the raft, just as we said we would. We hit the bottom of the drop, just as she said we would, and then our guide, our coach fell overboard.

Joyous rapture at our accomplishment turned to terror as we realized our commander was rapidly drifting away from our raft. I remember the deer in the headlights look on the faces of the guys in the boat, I remember people from the shore yelling something at us. I also remember one of guys on the boat throwing the “Hail Mary” towards our coach.

The “Hail Mary” was a joke our coach had shared with us a few hours ago when we were in calmer waters. She was talking about a throw bag, shaped like a football, you throw towards a person who is overboard hoping they can catch on to the rope and have a chance to get back into the raft. Fortunately, the throw was good (or the gods took mercy), because our guide was able to fight the undercurrent of the falls and get to the bag. Crisis one averted.

Crisis two had all of us gripped in fear. We were without our guide in the boat and were

drifting rapidly downstream with no experience to navigate the rest of the course. Something happened at that moment of crisis. Without a word being exchanged, we all realized the gravity of our predicament: we picked a direction, we paddled in unison and, like a single self-organized unit, put into practice everything we had learned over the last couple of hours to get to one of the shores. Having secured a stationary position on mother earth, we reeled our coach from the choppy waters into our raft.

*To see what
its like to raft
through
Husum Falls*
(See Video Link²)

Much of what happened next is a blur. The experience shadowed the rest of my journey down the river. I remember feelings of bitterness and abandonment. For if our guide was really good, really professional, she would not have flipped overboard in the first place. And I would not have had to fight for life and limb at the bottom of Husum falls. Our guide, on the other hand, was highly complimentary saying she was very proud of us and that we pulled it all together just like a great team would.

On our return car trip back to Seattle, my friends and I talked about our guide. Initially, we questioned her effectiveness and ability. But, as long road trips go, there are sober moments of reflection where the truth dawns upon you. We realized we probably had the best coach we could have ever asked for: she trained us on the basics of navigation, she trained us on working together, and she trained us on dealing with crisis. She prepared us enough that when it mattered most, we delivered. This realization that coaches are humans too and do err, told us her moment of coaching greatness was realized when she was not in the raft guiding us.

My rafting experience can probably be related to coaching software teams. However, I will not attempt to draw lengthy parallels. Having coached many software development teams, I tend to value my contribution by what a team does when I'm not with them over what the team does when I'm with them.

Links

1. http://en.wikipedia.org/wiki/International_Scale_of_River_Difficulty
2. <http://www.youtube.com/watch?v=5HF29q64Oas&feature=related>

The Coaching Mindset

A Conversation Between Dhaval Panchal and Michael de la Maza

Michael: What is the Agile coaching mindset?

Dhaval: There is the coaching mindset and then there's the Agile coaching mindset in terms of being a systemic coach, which is a school of coaching where you come in from a point of ignorance. Not being an expert in the field, where the coachee is requesting help. And you're using the strength of your ignorance to ask powerful questions or open up possibilities for the coachee. An Agile coaching mindset is a lot more specific around how might the coachee apply Agile values and principles to solving their challenges in delivering a product or a project.

This product or project could be work on themselves. And you can see the Agile values and principles tend to apply even at the level where you're discussing individual or personal growth and also for organizational growth, or more conventionally speaking, looking at how do I work with my team members in delivering this particular product. Agile coaching, to me, is a lot more specific around the application of the values and principles that are part of the Agile manifesto.

Michael: How do you differentiate between the coaching mindset and the consulting mindset?

Dhaval: A consultant typically comes in with an expertise that is arrived at with a lot of experience in a specific domain. The way I would look at it, a good consultant would be advising a lot based on what they would have done in a similar situation or what they think is possible here. Consultant is at some level is also responsible for generating a lot of ideas for the client. As opposed to coaching, where you are not there to provide them with all the ideas. And I will sound very trite in saying this, but coaching is about teaching the coachee how to fish as opposed to fish for them.

In other words, as a consultant, your approach is very specific to solving that given problem whereas a coach is looking beyond that one problem to see what can be done systemically about the person, their situation, and this person's approach to general problem solving, and how might I build a competency in this person so even though they are able to solve this one specific problem, they can apply this learning to solve future such problems. Which is very different from a consulting mindset because a consulting mindset requires that you have the consultant with you going forward, whereas the coaching mindset pretty much works against long-term employment. Right?

You're looking at how can I help or how can I coach my coachee into learning about themselves, into learning about investigating the problems so they can come up with their own solutions to their problems.

Michael: What happens and what do you do when a client hires you for coaching, and then when you get in the engagement, it's clear that they want consulting?

Dhaval: Generally speaking, before even I start engaging with the client, I try to get an understanding of outcome they are looking for. Do they think that my contribution is going to help them achieve that outcome? In other words, if they are talking about a very specific outcome and expectations from me are, "Help me solve this very specific problem or deliver this project." Then, I get a sense that you are probably talking a lot about consulting because our focus here is not about developing internal competency.

At that point, I try to make it very clear about what expectations they should have with us. The coaching versus consulting conversation. Now, if you look at the typical three questions, and I think Drucker talks about this, are you solving the right problem? Are you solving it the right way? Are you getting the right results? If you look at these three basic questions, the problem often revolves in and around these areas, which is: Are we solving it the right way? Are we getting the right results? And are we solving the right thing?

And typically, conversation about right way is where the conversation around coaching versus consulting comes into play. To go back to you original question, initially you may find that the clients are interested in coaching. And there comes a time when they're in a squeeze, or there's a time crunch, or budget issue and they don't feel like learning a way to solve that problem is going to help them in the short-term and they are looking for very specific advice. And at this point, I try to go and reiterate back to what we agreed to do, knowing that now we are shifting gears. I will be providing some very specific help through this particular problem area. And then, leaving the door open to going back to coaching.

'Cause, a disservice you can do to your client is solving all of their problems and pretending that you were coaching. 'Cause when you leave, they're now left without an ability to solve their own problems. They're unaware that by solving their problems, you are potentially avoiding a much bigger conversation around how might a coachee learn to solve this problem on his own.

Michael: And when you're evaluating the coaching mindset, say when you're helping to hire a Scrum Master or an Agile Coach, how do you do that? How do you know someone has the Agile coaching mindset?

Dhaval: I like looking for stories they tell. Simply based on an interview, it's very difficult to understand specifically whether this person has been a good coach or not. However, if you pay attention to the stories that people tell you about when they thought they were coaching, as opposed to when they were actually participating as a full-time team member, Scrum Master, or even a product owner in their teams, you will hear the choice of words and the actions that they take differs greatly.

Now, many people assume that they are coaching because they were able to install tools and practices for their clients. And you can quickly filter out people who take a stance that doing all the work implies they're coaching. In fact, one other trick I have often used is basically I ask them to tell me what was their contribution. And, if the amount of contribution they had was all the expertise. Then, I can tell very quickly that it was not about coaching but mostly about being an active contributor and not focused on the amplifying of contributions for other people in the team.

Michael: Right. Thank you so much.

Dhaval: Yeah, one more thing, I have asked this many times. It is not a hard and fast rule of mine but I do ask people to show me photos of their teams.

Michael: Oh. Interesting.

Dhaval: If you are a coach. And this is the general behavior pattern that I've noticed in good coaches, in coaches that I respect. On their camera, they tend to carry a lot of photographs of interesting moments where they found something interesting in their work. In the sense that they'll be taking a photo of a team at a very specific instance when the team realized that we should limit work in progress. They'll be taking photographs of team members or task boards in action.

Caoch's work was done almost a month ago, and now after the month, they see these seeds are growing. And they want to capture that. This to me speaks that they're really passionate about what they're doing and they're also interested in keeping the catalog or portfolio of work that they have done. Generally, you can recognize good craftsman by the portfolio of work they do. You know what I mean? If you look at the artistic people and think writers, think artists, they don't do just one drawing. They probably do thousands and afterwards, two or three make it to the world class stage.

You're looking for the body of work for someone as an Agile Coach, and to me, the best presentation of the body of work is photographs of the teams that they have worked with, photographs of the team space; perhaps a few blog articles or some kind of presentation done at a conference. These signal to me real passion for the craft as opposed to just me-to-coach jumping into space.

Cherie Silas, CEC



Cherie Silas is a Scrum Alliance, Certified Enterprise Coach (CEC) and an ICF Professional Certified Coach (PCC). Her background training includes training with CTI, ORSC, and Coaching 4 Today's Leaders. She has worked as a consultant assisting several Fortune 500 clients with large Agile transformations and serves as Director of Enterprise Agility in her current role.

Cherie is a member of the Scrum Alliance Certified Team Coach (CTC) review committee. She regularly volunteers as an executive coach with The Coaching Fellowship and The Center for Non-Profit Coaching. She also trains and mentors professional coaches through Coaching 4 Today's Leaders.

She has a strong desire to help people arrive at the place they define as success in both personal and professional life. Cherie's life mission that drives every interaction with every individual she encounters is simply this:

To leave you better than I found you with each encounter.

The Agile Manifesto — What It Means to Me

Original Source: <https://cheriesilas.wordpress.com/2013/09/15/agile-manifesto-what-it-means-to-me/>

The Agile Manifesto talks about uncovering better ways of developing software by doing it and helping others do it. The Manifesto goes on to state that through this work and helping others, four consistent values have developed. To me, these values go far beyond software development and set a platform for making decisions and forming thought processes. These values form the mindset of agility which spills over into every area of life. Because my mindset is one of agility, I can't help but take Agile out into the world beyond software development. Everyday, I work with people and see Agile changing mindsets and impacting lives for the better.

We value: Individuals and interactions over processes and tools

This value represents that understanding an Agile life is filled with humans! Humans are interesting, complex, intelligent, diverse, ever changing, and FUN! Processes are important and so are the tools that we use to get work done. But, when processes and tools become more valuable to us than the people who use those processes and tools they have overstepped their boundary. Processes and tools are created by people to solve problems, work more efficiently and to bring consistency. They should not be jails of solitary confinement where we get locked in and become slaves to the thing we created to help us! We cannot replace people with process and tools. When individuals interact with one another, creative ideas form, problems are solved, momentum is gained, new perspectives are shared, and growth occurs. People learn from interacting with each other. We become more aware of the world around us and more aware of ourselves when we interact with individuals of various types. When we take people out of the equation and rely on the processes and tools, our work suffers. Processes and tools are meant to assist people and should be used in this manner. They should never become a replacement for interacting with people. Text messaging is a prime example: texting is a tool that can be used for quick communication when direct conversations are impossible. But if we allow this to take away our ability to speak to and directly interact with individuals, we become a slave to the tool and it has more (negative) power than originally intended.

We Value: Working software over comprehensive documentation

To me, this value says: Let's don't just talk about it, let's do something about it! Let's build it! I can spend a lot of time writing a document that tells you every detail of what I can do and what I want. Or, I can write just enough to make sure you get an understanding of the direction we are heading and provide you with something you can touch and feel to see if it makes you happy. I don't want to waste your time or money and I don't want to waste mine either. So, lets build this thing together.

We Value: Customer collaboration over contract negotiation

I've got two choices when serving customers: 1) I can make them outline every detail of everything they will ever want from me and hold them to it rigidly (charging them for every slight shift); 2) OR, I can make them agree to create something great together, set some boundaries in a contract that protects us both, and start collaborating to ensure that we get to the finish line together!

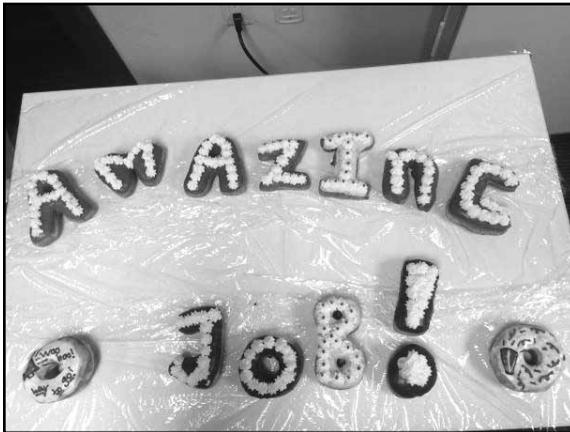
We Value: Responding to change over following a plan

Plans are good. They are needed. They are necessary. But, change is reality. Why do we pretend that we don't know that change will occur? People change. Circumstances change. Budgets change. Markets change. The world around us changes every single day. Instead of being ruled by a rigid plan that we know becomes obsolete and unrealistic just moments after it is created, let's plan to change. Plan in shorter periods of time that we are more likely to be able to predict for success instead of multiple months or even years down the road. Get feedback and don't be mad when the customer realizes they didn't know what they wanted until they saw what you provided. Be flattered that what you showed them generated enough interest and excitement they could see it become something great that met their needs and provided great value. Isn't that the end goal? If executing upon and controlling a plan is the primary goal, producing a valuable product that satisfies the customer must take a back seat to this objective. But, if customer satisfaction is the target — our plans must be flexible — Agile even!

Is More Really Better?

Original Source: <https://cheriesilas.wordpress.com/2016/07/26/is-more-really-better/>

One of my favorite things about being an Agile coach is connecting with the Agile community through conferences, meet-ups, and other networks. Because of these connections, I get



to interact with Agilists all over the world. Over the past few months I've noticed a concerning trend coming from the Scrum Master community. They are telling me with excitement, "I've finally worked myself up to two teams!" Some have said they are now working with three or four teams. The thing that concerns me is they seem to view spreading themselves across multiple teams as an accomplishment. I am hearing pride in "being busy" and "being able to handle more" and that tells me that we still have work to do. It tells me there may still be an anti-pattern running rampant in our Agile organizations telling us lies.

The belief that "the more I can handle and the busier I am the more valuable I am to the company" is left over from days when sustainable pace wasn't a part of the culture. The truth is that busy does not equal productive. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software. It is not to ensure that everyone is at least 100% (or more) utilized. Agile processes are supposed to promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely. When we are running at 100% (or more) of our capacity we cannot maintain that pace indefinitely. At some point, we burn out mentally, physically, and emotionally. We cannot afford to choose utilization over productivity. Our primary measure of progress should be working software, not how much more we can get done with fewer people. The efficiencies in an Agile organization don't come from piling more work on fewer people. They come from improving our technical practices, increasing automation, increasing quality, lowering technical debt, collaborating, and learning to continuously improve our processes. These things give us the ability to produce more without adding employees because we stop tripping over ourselves and can run along a clear path.

I read the following question from a user on stackoverflow.com:

"Does running your servers at 100% CPU usage cause any issues or **is it just good CPU utilization?** My servers have 8 physical cores constantly running at near 100% for "open hours"/10 hours per day. The program is architected to run on 8 threads – and it fully uses

them. Performance is good but the infrastructure guys are worrying about the “maxed out servers.” **I think it’s just good use of available resources. What’s the point of having lots of core if they are not all fully utilized?”**

The problem with this line of thinking is that when resources are fully utilized they don’t get more done. Contrarily, less gets done. They move slower, wait time increases, and so do errors. Here’s the response someone gave to this question:

“Almost without exception it causes issues, or will cause issues down the road (as demand grows). 100% CPU utilization on a web service server is not good. If your CPU utilization is at 100% it means that each time the server gets a new request there is a 100% chance that the work will have to wait some amount of time before the server gets started on it. The typical performance sweet spot is about 70%. Does that sound low? If so, remember that 70% utilization doesn’t mean that 30% of the CPU is being wasted. Instead, it means that 70% of the CPU’s capacity was used over a sample period. For CPU measurement metrics, a sample period is something like 2 seconds. During that 2 seconds the breakdown of that 70% is uneven. In other words, it may be something like 100% for 1 second and 40% in one second. For short bursts like that, 100% utilization is okay because we know that if a piece of work is delayed it is only for a brief period. (One that won’t make the human waiting upset.)”

I’m wondering, if we adhere to this rule with our hardware resources, why don’t we realize that the same rule applies to our human resources?

I’ve been in the position where I was a Scrum Master on one team doing an excellent job. I knew the pulse of my team and they were growing rapidly and performing better than ever. Then, I was given a second team. Sure, I had enough down time in my average week to handle facilitating Scrum events for two teams (in theory), but because I was toggling between two team rooms, I missed a lot on both. On sprint end/start days I felt very pressured. I ran from one retro to the next on and often couldn’t compile the improvement plan into a consumable format until two days later. I fell behind updating information radiators and had less time to think analytically through what was happening with each team. Over time I saw that both teams were maintaining, even growing some, but the rate of growth was slower than when I had only one team.

Then, something tragic happened. I was “doing such a great job” that I was asked to take on two more teams for a month to fill a hiring gap. I felt like a total failure. I had to choose which teams I was going to work with and leave the others stranded. I had no clue what was going on in any of the four teams because I wasn’t spending enough time with any of them to catch the important conversations. My teams all felt abandoned by me and had to pick up the slack felt by my absence. While in my manager’s eyes nothing fell to the ground (because my teams were mature enough to fill in the gaps without me), my teams felt all the pain and none of the benefit.

I learned a very powerful lesson through that experience. Being utilized at 100% (or more) capacity didn’t make me a super Scrum Master. It made me a terrible Scrum Master. On a ledger somewhere it may have looked like the company saved money by utilizing me to full capacity, but the impact of the hidden cost was much greater than the financial gain. We would have done better to allow the third and fourth teams to work without a Scrum Master for that month. Instead, we caused four teams to operate without a Scrum Master by spreading me too thin.

What message do we send as an organization when we tell our teams we expect them to plan

their sprints at 125% of their capacity? It's a message that says we do not value sustainable pace. What message do we send when we tell our employees we want them on multiple teams so we can fully utilize their capacity? It's a message that says we do not value sustainable pace. What message do I hear when Scrum Masters tell me proudly that they are working on multiple teams? I hear that they have forsaken the Agile principle of sustainable pace. I hear an anti-pattern. It makes me know that though we have come far we still have more work to do before becoming truly Agile.

Tips From the Trenches

Original Source: <https://cheriesilas.wordpress.com/2014/05/03/coaching-teams-tips-from-the-trenches/>

Earlier this week I met with a group of coaches of various experience levels from different backgrounds to talk about coaching teams. We discussed together our successes and failures in attempts to learn from one another. What follows is a list of the results of what we discovered together.

1. Create an environment where it is safe for people to fail

In order for teams and individuals to learn and grow they must be able to experience both success and failure. Most of us learn more from our mistakes and failures than we do from our successes. When we protect and buffer teams from failure we cripple them. When we give them all the answers to their problems and provide solutions for them we stunt their growth. As coaches, we have to step back and help teams have the courage to make decisions, investigate new ways of doing things, take risks, and explore areas they have feared to enter previously. They can't do that if we will not move out of their way and allow them space to succeed and fail. Failure is a learning experience. We can't always just take over when we see them struggling. We have to give them room to grow. We have to be confident enough in our ability as coaches to help teams navigate their way back up from failure to success that we have the courage to allow them to experience enough failure to grow and become higher performing. This doesn't mean that we should stand by and watch them walk head first off a cliff. As experts, we should know when to blow the whistle — but use the whistle sparingly only when it is no longer safe to fail.

2. Believe in people in ways that give them the courage to believe in themselves

As coaches we have to look beyond what we see standing in front of us today. We have to be able to look at what is in front of us today and see characteristics in people and in teams and roll those things forward weeks, months, and years ahead in our thinking in order to see the great things they have the potential to become. We can't think of it as what they might become. We have to see it as who they are. Coaches have the power to activate and unlock dormant gifts and talents in people by believing in them in ways that they can't even believe in themselves. People don't need someone to patronize them, they need someone who truly has vision for who they are and can articulate specifically what they see in them and why those things are powerful and amazing. They need a coach who can point out the simple yet amazing things they do and the impacts that those actions have on the team and on their career so people can have a light shining on the path that shows them what direction to walk.

3. Use The Language of Appreciation¹

Speaking to people in an encouraging language that tells them that they are valued and appreciated motivates teams and individuals and makes them want to move forward. It builds a solid connection and helps to form a trusting relationship with their coach because they see that the coach cares about them and believes in them. (See the other posts in this blog for more about the Language of Appreciation.)

4. Ask Powerful Questions

Serving as an expert has a place in coaching when it is time to teach. Becoming a mentor and walking hand in hand with people also has a place, but true coaching involves taking on a different role of allowing people to enter a place of self discovery. Asking powerful questions is an art that helps to facilitate this discovery process. Asking powerful questions can help people move outside the box of their normal thinking. Questions help them to develop their own conclusions and solve their own problems which means that they actually take ownership of the solutions and plans they make for their future. When people design their own futures instead of having those plans handed to them they are more likely to succeed at accomplishing the goals they set because they are motivated by their own ideas and empowered to make changes along the way to reach what they define as success.

5. Treat each team as individual and allow them to have their own culture/don't create a mirror image of yourself

Every team we coach has a different group of individuals in the makeup and should be encouraged to develop a culture based upon the individuals on the team. Even if the teams have a similar purpose they should have their own characteristics that are developed from within the team. I often view the multiple teams I coach like I view my multiple children. Each of them has their own unique character, strengths, and weaknesses. Each of them must be coached differently in order to become high performing. Each team must be assessed individually and the proper techniques must be applied that will help them grow. Making the mistake that we can duplicate the exact same methods, techniques, and cookie cutter process to every team we coach is harmful. We cannot expect that every team will look the same or to look like us — in fact, I dare say that if they do this is the sign of an immature coach. When I enter organizations and see teams that are identical I immediately think about the **cargo culting phenomenon²** where people do things that they see others doing because they think they will get some set of results. However, since they don't really understand the underlying reasons why the first person took those actions the repeat of the behavior adds no value.

6. Don't get in the middle of conflict – force them to storm instead

Teams need to have constructive and healthy conflict. Sometimes the conflict turns unhealthy and people don't want to deal with it properly. There is a very real temptation to try to solve the problems of the team by getting in the middle and handling it for them. Bad idea. As a coach it is our job to teach people healthy ways to resolve conflict so it is better to help individuals form a plan for confronting and dealing with conflict or to create a way to surface the conflict with the team. In order for teams to become high performing they must first go through the process of forming, storming, and norming. Unfortunately, too many teams never really storm because they never learn to have healthy conflict. The elephant stays in the room and everyone just walks around him. Teaching individuals and teams to

address the elephant together using appropriate and safe communication styles, healthy conflict resolution techniques, and problem solving skills serves a better longer term purpose than getting in the middle as a go between to make today more peaceful.

Links

1. <https://cherie-silas.wordpress.com/2014/01/14/the-language-of-appreciation-in-the-workplace/>
2. <http://www.damninteresting.com/john-frum-and-the-cargo-cults/>

What Does Sustainable Pace Really Mean?

Original Source: <https://cheriesilas.wordpress.com/2016/07/28/what-does-sustainable-pace-really-mean/>



Agile values and principles are the core foundation by which Agile organizations operate and make decisions. Everything we do is based in these values. With that being said, viewing every principle through a holistic perspective is absolutely necessary. Every word in the principles we live by has value and impact. So, when we reduce a principle to a three word summary, I believe we do ourselves a disservice. This practice often results in focusing on part of the principle without the balance of the other side. Through this oversight, we

inadvertently create environments where there is unbalance that leaves people frustrated and confused. They begin to believe that Agile is the problem. But the real problem is our failure to completely embrace Agile values and principles, and we settle for anti-patterns instead.

Today, I'd like to take a deeper look into Agile Principle #8 which states: Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

People often refer to this principle as the sustainable pace principle. The most common description given of how we practice this principle is that the development team should not be expected to do more work than they can complete in a normal business day. We don't want people working 70 hours a week because they are forced to do more work than is possible during a normal work week. Working at that pace is something a team may be able to do for a sprint or two, but they cannot work at that pace indefinitely. When people are tired and overworked they make more mistakes and it actually slows down their ability to produce work. It also impacts motivation. When people are overworked and have no work/life, balance motivation dwindles.

But there's another part to this principle that I don't hear quoted as often. It's the part that talks about the constant pace which the sponsors, developers, and users should be able to maintain. This is about the consistency of our delivery, sometimes referred to as

predictability. Developers should be able to trust that sponsors and users will allow them to work at a sustainable pace. In return, sponsors and users should be able to trust that developers will consistently provide a continuous stream of valuable software. The team has a responsibility to be transparent with the sponsors and users regarding how much work they can complete in a certain timeframe. They also have a responsibility to be transparent when the forecast must be changed along the way due to new information or unforeseen problems. This gives the sponsors and users the ability to communicate and make decisions regarding the impact of the forecast change.

How does this impact the way the team conducts planning and communicates their forecast? Teams should plan for as much as they can realistically complete and then communicate that forecast. Teams should strive to complete 100% of their forecast every sprint. If something happens to prevent the completion of the forecast, they should communicate such as soon as feasible to stakeholders so they know what to expect.

Should teams forecast 125% of what they believe they can realistically complete and be happy if 80% of the work gets finished? No. Why? First, because it sets unrealistic stakeholder expectations to communicate more work than the team can realistically expect to finish. Second, because it contributes to a lack of trust between the stakeholder and the team when the team keeps promising work they consistently don't deliver. Third, because the extra time planning and tasking stories that aren't likely to be worked creates waste and adds unnecessary time to the planning process.

Then what do we do with "stretch" stories? It is my belief that "stretch" stories are not a part of the forecast. Plan and communicate what you believe you can complete. If the backlog is groomed properly, it will always have at least 1-2 sprints worth of work in "ready" state. So, if the team runs out of work, they can always agree to pull in another story. The solutioning and tasking for that story can take place when the decision to pull it in happens.

If the team consistently gets 100% for 3-5 sprints, stretch yourself and bring a few more points into your sprint forecast. It may take you a couple of sprints to get to 100% again, but it will stretch your ability to produce work and push you to incorporate practices like automation in order to move faster.

There should be an understanding that no team will always complete 100% of the work forecasted. This is another part of the concept of trust and transparency. Stakeholders and customers trust that developers will always strive to complete 100% of the forecasted work. Developers trust that when something happens and they can't deliver 100% and communicate openly to stakeholders and users there will be grace and understanding extended.

Chris Steele, CEC



Chris Steele is an enterprise coach (CEC) and an organizational change specialist. Working with companies ranging from startups to global Fortune companies on four continents has provided him with a wide range of experiences and keen insights into the problems and solutions that companies find when embracing Agility, and on how to help leaders discover ways to overcome their organizational inertia to enact real transformation. He has significant experience as an executive coach, speaker, workshop facilitator, and process methodology consultant.

Chris is a former software developer and architect who has a keen interest in technical excellence and in raising the bar to hyperproductive high quality delivery in every organization he works with, including global mentoring and training programs. And Chris also works with software development leaders in both a training and coaching role, working both top-down and bottom-up to fight against “bad Agile” and to facilitate the discovery and implementation of simple solutions to complex problems.

Now residing in the suburbs of Chicago, Chris leverages his deep expertise in process improvement, creative facilitation, and the passion that he discovered back in 2002 when he started his Agile journey, and is currently focused on enabling frameworks, practices, and tools for unified global delivery at scale in large organizations, and in growing local leaders in low-cost geographies. He holds a Master’s from the University of Liverpool.

The Successful Coach

Like many others who will be contributing to this collection, I've been fortunate: I've had the opportunity to work with companies all over the world, ranging from less than 100-person startups to global Fortune enterprises with more than 250,000 employees. Each client has pursued their initiative slightly differently (enterprise coaches, swarms of consultants, centers of excellence, "in with the new guard," and many more paths), but they have all had roughly the same goal in mind — change. They've recognized a need to alter the way that they work. Or, at least, they're unhappy with the current results they are getting and believe the way they work may be one of multiple root causes for those results. The outcomes have varied significantly with each company, and I want to spend a few paragraphs reflecting on why that is so often the case and how a coach contributes to a proper outcome.

Change, as the old saying goes, is hard. In fact, I believe the largest success factor in any kind of enterprise transformation is a willingness to embrace change at all levels starting from the top. First and foremost, change is NOT other people. It's everyone. Leadership must lead, as it turns out, and this is often a far different exercise than the "mandate and measure" strategy that is so often used in operational endeavors. One of the most difficult things a coach does is to carefully, honestly, and respectfully guide leaders to recognize failures of leadership and their impact, and to help those leaders arrive at a better set of behaviors and thinking. Everyone has a certain amount of fragility in their ego, and company builders are of course no exception. A CEO hearing "you're the reason this isn't working" from a coach, when the CEO's built a company from an idea into an industry-leading behemoth, is not likely to respond cordially, and the coach's assessment introduces some real cognitive dissonance as the evidence of the CEO's success and good decision-making is literally all around him. The coach who cannot overcome this hurdle will fail.

This goes down the ladder as well. There's a lot of work that needs to be done in making the case for change clear, creating a sense of urgency, and making it relevant to other people. By that last part, I mean, what does an entry-level employee care about the ability of the board of directors to pay out dividends to shareholders? Probably not much, so that's not the appropriate context(although it may be the scorecard at a certain level of executives). Coaches challenge, guide, interview, and leverage a history that allows them to help people discover for themselves why change is important and what role they play in the process.

While change is inherently hard, it's even harder to sustain. Change requires a sense of urgency; a sense of importance. It is vital to keep up what can be an emotionally and mentally challenging transformation process for the length of time required. It is especially important to understand the process correctly and to frame activities and results in a way that inspires further action. To create an impetus for real change, there needs to be an admission of failure. After all, if the way things were being done were working, there'd be no reason to change at all. Still, nobody can feel like they are failing forever. Such feelings can demotivate and cause the dreaded "fear, uncertainty and doubt" that can destroy an organization like

a cancer. This is where measurement CAN be important — a good transformation comes about with careful, frequent self-reflection in which in a very public way wins and losses are discussed, organizational learning happens, and new goals and activities are planned.

This is a kind of publicity that is often counter-cultural. Many organizations simply cannot tolerate this kind of communication, this kind of inspection and adaptation. There are a lot of reasons for this which I won't go into here, but at some point cultural roadblocks arise, often in the form of people who are either not onboard or actively trying to prevent change. The people who are most likely to be impediments to change are often those who: a) have been wildly successful in the past; b) control large budgets and headcount; c) have staked their career on climbing the ladder of the organization. They understood the rules of the game, played it well and now their worldview is getting shaken.

Successful organizations will come to understand that future outcome is more important than historical behavior or achievements. Some people (who are great in many organizations!) will find their way to an exit — hopefully guided in a mutually positive, compassionate way by their peers and supervisors. This can really only be done successfully when the change comes from internal agents and champions. A consultant might say “Joe really is getting in the way, and he has the potential to derail the whole thing. We need to x” — where “x” might be educating Joe, escalating the issue, or trying to remove him altogether. That kind of approach doesn't work — it creates an “us vs them” mentality. Instead, a coach helps ask questions that guide people's thinking into discovering for themselves the risk that Joe poses and coming up with action plans of their own to try to solve it. This creates a culture of learning and accountability, and will sustain change far better in the long run. This is just one of the many examples that demonstrate how coaches endeavor to put themselves out of a job in every organization with which they work.

So what makes a coach successful? Obviously, subject matter expertise is good to have. However, you'll notice that subject matter expertise is almost secondary. As we learn and grow in our careers, the toolbox we possess becomes full of different tools for different jobs. The software developer who becomes an architect may become less fluent in coding, but still understands coding and can assist others. Her highest-value skills, though, change and she must pick up new approaches, thought patterns, and abilities that she may have never needed before. It is the same in coaching. Successful coaches have an approach, a toolkit, a collection of frameworks that can be utilized to focus conversation and thinking. Coaches can help establish the behaviors in individuals and organizations that create real change. The way one coaches an executive vs. a team or vs. a middle manager may be very different. Handling change at the division level or the organizational level is different.

That's not all that a good coach needs, though. Understanding the business' value chain and the patterns that successful organizations in similar lines have utilized is another key. Rather than “simply” being the subject matter expert who can say how a thing should get done, the coach understands how to make other people discover for themselves what should be done. A coach helps a business identify and overcome their own roadblocks (both known and unknown). Coaches have enough experience and realism to say “x is a great goal, but here are a few stories and/or thoughts that may lead you to consider doing some other things first to make that more successful.” Coaches are consistent; they help organizations bring process and transparency to chaos, and they care deeply about people. Coaches are patient but tough; establishing working agreements and holding people to them for everyone's benefit. Coaches know when to cut the cord and let people make their own mistakes. And, coaches practice what they preach: coaches are always learning.

Sometimes the hardest part is letting go. It's a joy to see an organization reach their goals and embrace change, establish good patterns and behaviors, and achieve the outcomes they are hoping for. It's harder to identify when it just isn't working and to make the decision to end the coaching relationship. And regardless of how it ends, the coach who becomes invested in the people he works with has the same bittersweet feeling that comes any time one leaves a job. Except a coach might be ending relationships at three different places at the same time. Coaching is rewarding, challenging and never-ending. Coaching is, as we say, a hard way to make an easy living — but it's worth it.

Mark Summers, CEC, CST

Mark Summers is a Certified Scrum Trainer and Certified Enterprise Coach with Beliminal, helping individuals, teams and organisations with their Agile and Lean journey, since 2006. Mark is committed to help support the development of environments where people can take pleasure from their work, delighting customers with quality products.

Within the Agile community, Mark is a leader on Agile Coaching; facilitating coaching clinics at conferences, organising Scrum Coaching Retreats across Europe, and mentoring other coaches. Mark was a recent keynote speaker on the role of coaching in Agile organisations at the Scrum Gathering in Munich.

Mark is also a professionally-trained coach in business and personal coaching, accredited to Professional Certified Coach by the International Coaching Federation.

How Do I Use My Agile Coach?

This article is based on a talk that I gave as part of the Keynote mash-up at the Scrum Gathering in Munich 2016.

Transforming the world of work is about investing in people — specifically about building a coaching capability to support people. So when we bring outside expertise into our organisations such as an Agile Coach, how do we best use them?

So why the coaching focus? Well, let's start by taking a look at what professional coaching is.

What is coaching?

Unfortunately, not everyone agrees what coaching is; there are many definitions of coaching, but at the heart of it, every great coach believes:

- People are different from each other — so we must treat them as the great individuals they are (so they aren't resources).
- People always do the best they can given their situation. So as a coach, when an organisation calls me in to deal with a performance issue ("my team needs fixing"), I never treat it as a performance issue. It is always because of some waste, organizational policy, or the way people are being managed.

In fact, coaching is not about fixing people. We can all change, we all have choices about who we are. And, we all are resourceful enough to make changes should we choose to do so.

A coach believes an individual is the best expert in their own lives. A coach trusts that if he/she nurture and support them, then the best way forward will emerge from that individual. A coach must trust the person or team they coach.

Scrum and Management

Scrum is a trust framework with the intention everyone outside of the team trusts the team. Scrum coaches are experts in their work to deliver the best solution. The idea being that management's role is to nurture and support those teams. Not by focusing on performance, but by believing if they can create the right environment, the teams can excel.

So, there is a lot of synergy between a coaching approach and the role of management in a Scrum environment. Yet, most Scrum implementations I see are used to micromanage the team. Scrum is being used as a management tool that seems to stifle creativity and innovation rather than empowering the teams and setting them free.

Why is this I hear you cry? Why does this evil afflict our Scrum adoptions?

"Bob, welcome to our organisation." (Bob is an Agile Coach.) "We are so excited to be adopting Agile, and glad you came on board. So the team is over there. Go make them Agile."

"Wait," says Bob, "I will need to work with you as well."

“Why?” responds the manager, “it’s the team that needs to change, not me.”

Guess what, your team is not the problem — they are great as they are.

It’s this mindset: Scrum is what the team does. Scrum is a great, but we the management are not going to change anything. This mindset is common, but it dooms the adoption of Scrum to being superficial and brittle.

Agile Adoption in Large Traditional Organisations

In the UK over that last few years, larger more traditional organisations have started to jump on the Agile bandwagon. Often, Agile is now brought in by senior leadership — some of them even know what it is. However, they all like the sound of the benefits we claim are available to them: hyper-productive teams, delivering more for less, delighting customers. Let’s shoot that silver bullet.

Of course, Agile is not a silver bullet; it’s a cultural shift that is, in fact, part of a wider paradigm shift to decentralize control. This is a cultural shift that takes a lot of intentional effort and a lot of discipline.

Some of these large enterprise organisations take the approach of hiring a lot of Agile coaches (“We need a 100 Agile coaches”). Maybe they do, maybe they don’t. But often these Agile coaches are expected to go work with teams. And of course, we need to know the coach is adding value. So on a weekly basis, Agile coaches must fill in a report and answer questions like:

- Does your team have an information radiator?
- Does your team know who their Product Owner is?
- Does your team have retrospectives?

Now, the team might be very successful with all this support from an experienced coach. I remember one team I coached: when I started with them I was nervous because one of the team members had a book on how to write Assembler (and most of the product was written in Cobol). The output was going to be a fix to the charging model and a batch process that would give money back to all those who had been overcharged (“How is this going to be an Agile?” I thought). But I trusted the team and they worked it out. They didn’t have a Product Backlog because they didn’t know what they needed to do. But using an incremental and iterative approach, the team fixed stuff and found more issues. The level of quality they got was amazing — no bugs! They had a cross-functional team which included business people who understood the policies. Also, the project was completed in 2 months less time than similar projects. Success: happy customers, happy team, no production defects. I asked the team if they would use Scrum on their next project when I was gone. They said, no, their manager wouldn’t let them.

I went to see the manager. He told me Scrum wasn’t going to be appropriate on the next project as it was too complex. At this point, he wasn’t open to hearing that if you have uncertainty, Scrum might be exactly the right approach. The other thing I sensed was fear: the team, with my support, had been in control (the manager was not involved in the process). He was losing his control. If this thing spread, what would he do? He might need to change. He could count on two hands the number of years until his retirement. He takes home a healthy reward, why should he change?

Some people might call this the frozen middle.

I had succeeded as a coach to support the delivery of this first project, but I had failed as a coach to build sustainable transformational change. I had failed to show middle-management how to support the team, how to coach teams once I was gone. It's not the only time I have failed as an Agile Coach. We get short-term success just growing great Scrum Teams, and that feels good.

I had helped create a team culture, but I had done nothing to address the organizational culture that is usually embedded in the middle-management.

Now I have learned. I now spend more of my time with executives and management. Although, recently one manager running the Agile Transition said to me, "Mark, you are spending all of your time supporting the leadership team. You must be frustrated because you are an Agile Coach. Surely you want to go and work with teams?"

Well, that might be fun, but it is not where the real challenge lies.

Our Mission

As Agile coaches, it's all about where we choose to focus from this point forward. We need to address our efforts to the frozen middle. In fact, the frozen middle needs to be replaced with a coaching capability that nurtures our teams. Given this environment, our teams will look after themselves.

We need to go beyond the basic training courses. We need to go beyond working with the teams and we need to go beyond Scrum. We need to have courage. Things may get harder for us, but if we don't change, then we will not grow and we will fail to transform the world of work. Our mission is not to get as many teams as possible to adopt Scrum, especially evil Scrum. Our mission, should we choose to accept it, is to transform the world of work. And transforming the world of work is a culture change, not a process change.

As the role of management moves from commanding, controlling, monitoring and decision-making to one of growing individuals, teams and continuously improving the organisation,. more than anything else, the Agile transformation is about building the capability within middle management to coach teams once the Agile Coach leaves.

For those of you in organisations, when you get Agile coaches, consultants and trainers, yes, you need to make sure teams know what Scrum is. But then, trust the team. After all, the problem is not the team — not if we trust the team.

So, to all of you out there representing organisations: invest in your people and build a coaching capability. Adopting Scrum at the team level, without addressing your frozen middle, will mean you won't really have changed anything. And once your coach is gone, and a new CEO brought in, your investment will be gone.

In Summary

In a transformed world of work a key capability of management will be coaching. It is also an essential skill for the Agile transformation. Therefore, let's focus our Agile coaches on helping to build this capability.

Chris Waggoner, CEC



Chris Waggoner is a CEC (Certified Enterprise Coach) and Managing Consultant at SolutionsIQ who specializes in Enterprise Agile Transformation. He has been an Agile change agent for 7 years for multiple clients within the United States and Great Britain. Chris blends his 30-plus years of experience in Information Technology and Consulting with his more recent Agile experience to guide organizations through the minefields of organizational change and impediment removal. Chris is pragmatic in his approach to Agile adoption, understanding that each individual, development team, organization, and executive team adopt Agile and change at different paces and rhythms. Knowing that nothing speeds change like success, Chris has learned and developed many facilitation techniques that help organizations see the potential productivity gains of Agile adoption early in their adoption journey.

When It's No Longer Relative

Original Source: <https://www.linkedin.com/pulse/when-its-longer-relative-chris-waggoner>

Estimation of time and effort until you're done is one of those things every inquiring manager wants to know. In an effort to provide "accurate" estimates, companies and people spend countless hours generating lists of tasks and debating the duration of each task. When it comes to developing software, these estimates are rarely correct and often grossly wrong. The estimates are wrong because in building software, we are often building something that has never been built before. When we find ourselves being pressured to estimate something we've never done, we tend to SWAG (PMI for Sophisticated Wild Ass Guess) the answer in man-days and/or hours without understanding or knowing all the potential variables in play. Herein lies the problem with traditional software estimation methodologies: within a short amount of time spent there are diminishing returns on effort versus value. That is to say, early in the traditional process the value of spending more time estimating doesn't provide a higher certainty of when we will be done.

In the Agile world we are always looking for ways to reduce and prevent waste. One area of waste we focus on is estimation. We want to reduce the time spent on the diminishing returns of effort versus value in traditional software estimation. In Agile, we do this using relative estimation or relative sizing. Relative sizing is simple for us primates because our brains come pre-wired with the ability to do it accurately. We easily see small banana, medium banana, large banana and, finally, extra-large banana. We quickly size and sort eggs, post-it notes, boats, and an almost unlimited number of things we come in contact with in our daily life. We can also size and sort abstract things such as the small problem I can handle now, the medium problem I need help with, and the large problem I have no clue how begin to resolve. The hard-wired human ability to size and sort is at the heart of Agile relative estimation techniques.

When we train teams on relative sizing we ask them to size and sort their software requirements (or stories) within their backlog. These teams may use something as simple as T-Shirt sizing (S, M, L, XL) or they may use a more sophisticated Fibonacci scale (i.e., 1,2,3,5,8,13 ...). With stories in their backlog sized, we take teams into Sprint Planning. At some point during Sprint Planning training, we explain the importance of the commitment, the number of stories or story points they are committing to complete this sprint, and then ask them to commit to a number of story points they select. It is normal at this point for a new team to fall silent. Since they've never sprinted before, they have no clue how many points they should commit to. To help them with their first commitment we use a training approach. Part one of the approach is for the team to calculate their capacity:

- Each team member determines how many hours of capacity they have available during the sprint;
- The team determines how much slack they need in sprint capacity (normally 20%-30% for administrative work and other non-sprint related activities).

- Available team capacity = gross team capacity minus slack.

Part two of the training approach is to fit committed stories within available capacity:

- Team tasks out stories;
- Team estimates hours per task;
- Team totals number of hours per story;
- Team adds stories to their commitment until their available capacity is full;
- Team commits to story points that sum to their level of available capacity.

Did you notice what we just did?

We just created waste in estimating. We left behind the basic principle of Relative Sizing. We asked for a SWAG on the task created. We just sent this new team down the slippery slope of adopting bad estimation habits. We have invited and enabled any manager watching the training process to hold teams to an hourly estimate. Our estimates are **No Longer Relative!**

This training approach, in my experience, seems to be universal among trainers, coaches and Scrum Masters and here's the problem: we don't tell teams this is a temporary tool designed to help them learn how to commit until they've found their velocity after 5 to 6 sprints. We don't take this tool away, we just leave it there for the team to incorporate into their culture. We don't take the time to explain we are purposely breaking a relative sizing, why we're breaking the relative sizing, and why it's not a good idea to continue breaking relative sizing. We just leave it there and let it fester into dysfunctional behaviors.

Perhaps you've run across some of these behaviors? Consider some common things that I hear when working on this issue with teams:

- **Team Member: "We assign 2 hours to every task."** — Team has unconsciously realized that the hours are meaningless. They are just putting hours on task because they were told to.
- **Product Owner: "The hourly burn-down chart is how I know if the team will complete the sprint"** — The hours are false, they were SWAG-ed; therefore the burn-down is false. Ever notice how hourly burn-downs take a nose dive on the last day of the sprint? The only burn-down chart that has any real consistent value is a story burn-down chart. What value do estimated hours provide if stories aren't done?
- **Team: "We should get credit for the hours we completed even though the story is not done."** — I've found that this discussion leads to hourly estimation driving a misunderstanding of velocity.
- **Manager: "Every day the team needs to adjust their hourly task estimates to reflect reality."** — Really? How much waste can a manager create? To what purpose are we readjusting and reconciling hours estimated versus actual hours?
- **Inexperienced Scrum Master: "We assign story points to a story during sprint planning based on the number of hours tasked with the story. We have a sizing chart posted in our team room based upon a sliding hours estimated scale."** — I fell out of my chair when I heard this one.

I've quit using this training approach because of the dysfunctional behavior it creates. The approach is **No Longer Relative**. I don't want any manager within earshot to hear the words "hourly estimate." I've even gone as far as to tell a team to task stories only if it helps them accomplish their work. Tasking is a good idea because it helps teams organize. I suggest tasking, but don't prescribe it. I let the team decide if tasking helps them or not.

Velocity is going to be what velocity is going to be. So, let the first time Scrum team pull a

commitment number out of thin air. Tell them we're not going to take your commitment seriously until sprint 5 or 6. Don't ask them to put hours on task. If you do, you will no longer be relative.

BUSINESS

