

The Tail that Wags the Dogma

Rhea Stadick

rhea.d.stadick@intel.com

Abstract

It is a common pattern for quality advocates or advocates of any kind really (Agile, Waterfall, etc.) to start off banging a drum with almost religious zealotry. "This is the way to achieve quality!" "Follow me or be cast aside!" I have certainly done this and many of the change agents and forward thinkers I respect in the industry have struggled with this as well in their careers. However, riding a train of dogmatic viewpoints and practices has, in many cases, ruined the original efforts of change agents and more tragically, created bad patterns of development. This paper and presentation will explore how we can clarify our intentions and work towards creating an environment where the people involved are empowered to think and experiment but still march in the same general direction. As product development becomes more and more complex, different approaches are needed to get to the right outcome and we can no longer afford to apply rigid methods that worked for a simpler environment. This paper will help the audience think critically about where they are being too dogmatic in their quality efforts while introducing some new ways of thinking around complex adaptive systems and complex product development.

Biography

Rhea Stadick is an Organizational Coach at Intel, Corp. She has spent the last eight years in software quality and development of engineering teams. Today she helps organizations across her company develop cultures and competencies to create thriving work environments that support excellence in product development. She received her B.S. in Computer Science from Oregon State University and M.B.A. from Willamette University. For the past several years she has organized the Rose City Software Process Improvement Network (SPIN) in the Portland-metro area that gathers professionals in the area to learn and network.

1. Introduction

Part of my early career began in a quality advocacy role. In my organization I was immersed in a mindset that “we needed to control quality and that if we only used our expertise we could devise great processes that would lead to the high quality goals we sought.” Unfortunately, the people using the processes got in the way (“why do they keep doing this incorrectly?”) and reality hit (“why is schedule pressure always trumping quality investment?”), among other exasperating issues. I began to see that despite everyone’s best intentions, the process and practices we put in place just weren’t working even though they were well accepted as best practice. Something else was happening in the environment that we weren’t acknowledging.

This paper describes my continuing journey of growing my understanding of systems to create an environment that allows people and business to thrive. While I don’t prescribe a recipe or a new profound way of working, it may help you think about how you are approaching your work and change agency today. Perhaps it’s great, perhaps you may take another approach next time. The important thing is to continue to learn and not take your current state as the only way you’ll ever be or approach your job. We all could do better to not just use recipes but to really learn how to cook in any context.

2. View from the Ivory Tower or “You’re Not Trying Hard Enough”

The “beat the drum” management technique of telling everyone they just need to work harder should be relegated to one of those de-motivational posters. You don’t create high quality products through brute force, so why do we think putting in more metrics, criteria, and process will help? Ironically, despite the overwhelming power of reality, we as humans have a tendency to rationalize away that reality to fit the story we’ve created in our minds. I’ve done this with quality processes, blindly attributing lack of adherence to the process to lack of understanding or discipline. There have been countless case studies, thrown at MBA students, demonstrating companies that, despite all evidence to the contrary, chugged along with the same approach and mentality in product development right up and over the cliff of oblivion. Remember DEC?

Historically, quality methods have taken the form of prescribed processes to enforce control of a system to produce the desired result. While leaving some room for adaptation, it was understood that there was a defined way to produce high quality products. Further, the process to this day is often king. Unfortunately, this is often too much of an ivory tower view where from far away, people look like rational parts of the system. However, people and organizations rarely, if ever, operate in a rational way. Our approach has to accommodate “imperfections” (aka, people being human) and not try to force a rigorous and unbending process on them. Otherwise, we risk taking the humanity out of a very human system.

I see this exemplified by many of us orthodoxly holding to the pseudo-waterfall, stage gate life cycle of the company (driving process based on it, collecting metrics supporting it) despite evidence that no one is really following it as it was intended. One of the stages didn’t go right but we still pretend that we can continue to use the same process even though it is based on the assumption that we cleanly exit one stage in order to enter the next stage? Where’s the emergency lever here?

Another case in point is collaboration. We desperately want teams to communicate more across our products but our organizational structure has been created based on the current life cycle of

the company that, by nature, creates silos. When we take certain portions of the process for granted and consider them unchangeable, it's very hard to see another way of working outside of just hammering on ourselves to get better at the process.

At this point, blame becomes a favorite tool. "If only marketing had gathered the requirements correctly we wouldn't be in this mess." Fast forward several months of working nonstop to "do whatever it takes" to get the product out the door. We hold a retrospective to talk about the fact that if we had only followed the process we would have been fine. Subsequently, we get a little bit better at following the process but still run into many of the same issues as before.

This happens at all levels of organizations and to the best of us. I'd advise right now if there's anything you're trying to change and you have a little voice telling you, "I wish they'd just listen and actually try this," and that voice has been saying this for a few months then QUESTION THAT VOICE. Perhaps there's something going on that doesn't fit that story line and you need to quiet that blaming/justifying voice if only to allow some room for reflection. One way to get this reflection is to get support. Ask someone outside of your domain to play devil's advocate for you and just listen to the questions they raise about how you're approaching your work. Don't be afraid to use your customers as a source for good feedback.

2.1 Herding Cats

"Nothing pleases people more than to go on thinking what they have always thought, and at the same time imagine that they are thinking something new and daring; it combines the advantage of security and the delight of adventure."

– T.S. Eliot

But what else could be going on? At one point in my journey, I discovered methods to enable the quality change that I wanted to make. I found Agile (queue the inspiring music) and saw that we had to look not only at the mechanisms within the life cycle where we worked but at the life cycle itself, the people (surprise!), and the principles we followed. The system level approach was revolutionary in my mind and corrected the many problems I had found in my past approaches to driving product quality. One of the key things that I had been missing was that not understanding the whole system meant I was optimizing for only a few parts of the system. However, these parts were affected and changed based on other operators in the system, of which I had no visibility and frankly wasn't paying any attention to. There is no way to really way to get the sustained outcome we want unless we start to understand the system in which we're working.

Unfortunately, I still approached it with the same black and white stick – "follow this Scrum process or else!" "Agile is the only way!" I spent more time than I care to admit brute forcing this method on our team with a religious zealotry. I provided examples of teams and data to support why Agile was better. While this made me feel better, it didn't go a long way toward helping change the situation or people's perception. In retrospect, I was using the same dogmatic approach but substituting "quality" for "Agile." I also focused much more on something that was easy (the Scrum framework) than the harder cultural and mindset change that Agile endorses. This is not to say that we didn't see enormous improvements in quality, team health, execution, and engineering by making work transparent and building in mechanisms to reflect and take action as a team. But by favoring the process as a mechanism for change versus focus on the ultimate goal of being an Agile organization, it just got us to be really good at one thing. Meanwhile, we weren't looking around at what other ways we could approach our work in order to do even better.

Over the course of a couple years, through hard work and dedication, we were able to get good at getting things done in the sprint. However, we saw some anomalies such as parts of the code

being hard to change, validation getting squeezed at the end of the sprint, and the team not becoming the dynamic, happy group of professionals that Agile promised. In many ways we had plateaued in our improvement efforts and weren't seeing a better way to work other than to try harder. Luckily, we hired an engineer with a background in Extreme Programming (XP) that lead and inspired the team to use Agile technical practices. The engineering team took a big step forward and moved to a much more Agile state in only a few months. Where I had been coaching the team on a particular methodology, our new hire brought in new practices and was able to take the team to the next level. Having someone with a new perspective come in to coach the team was invaluable.

Given how much the team evolved in such a short period of time, I realized from this experience that I was just using one tool and acting as if it would be the solution to all problems. Yes, Scrum was my hammer and everything in our development was a nail ("if only we just did Scrum correctly it would be fine!"). I was fixing my mindset in a way that didn't challenge trying new approaches to the detriment of increased improvement. I had put so much emphasis on the process, I didn't step back and focus on the reason we were doing this process and what our ultimate goals were. Looking back at that experience, I would tell myself and my team to always look at the why, look outwards, experiment, and crave continuous improvement.

If this sounds familiar, then know that you're definitely not alone. It's very easy to adopt new tools versus new behaviors and approaches. Every day now, I challenge myself to see if I'm really approaching my work differently and if I'm trying something new or if I'm carrying along my old mindset baggage. The quote at the beginning of this section helps keep me humble! However, it's best to ask those on the outside. Ask your co-workers or your clients. My team actually goes out to other corporations and speaks to people in a similar line of work as ours. This helps us get fresh ideas and challenge our current approaches. We also bring outside change agents in to take a look at what we're doing and give us an outsider's perspective.

3. Peeling the Onion or “Stories We Tell Ourselves When Things aren't Working”

Our engineering teams were doing exceptionally well but when one part of the process broke down, such as the Product Owner not fulfilling their role, we found ourselves in fairly serious circumstances. Many had a sinking suspicion that no one was really challenging what was valuable or not on the project. In fact, how did we really know the product was even worth doing other than our upper management saying that it was valuable? The reality was that we had lack of organizational support or understanding for this role and subsequently did not put people into this position that would be successful. Instead of accepting the reality that no product owner support was coming, we dug our heels in the dirt. We kept our focus on our 2-week scrums and heavily rejected new work coming into the sprint because we assumed the Product Owner was not prepared. Had we created a monster? We had fixed our mindset that the process was king and that people just needed to try harder.

Using Christopher Avery's Responsibility ProcessTM as a guide, we were definitely in the blame and justify space. What we weren't doing was taking responsibility. The story we were fixed in was that we were the victims and, if other people did their job right, everything would be fine. We had stopped looking at why we were doing the things we were doing and acting from the present reality versus an ideal one we sought. Agile tells us that we should always be focused on creating business value. Thus, we should have stopped and asked ourselves, "if we don't really have a focused product owner, aren't we just guessing about whether we're creating value?"

Unfortunately, we went along with “the program” and didn’t challenge what was happening. I wish I could say that this story ended well but after the team completed the entire year and a half project and right before launch, the business group pulled the plug and decided not to ship it externally.

Earlier in the project, Scrum was doing its job as a tool by telling us that something was wrong and sending up warning signs. Unfortunately, instead of listening, we hunkered down and focused on the development, blaming the rest of the system for the problems instead of challenging the organization to really make a choice about whether this was a valuable product or not. To make this kind of challenge would not have been easy; challenging the status quo is never a walk in the park. But, we have to ask ourselves: isn’t questioning the validity of the product better than wasting months of our time on work that doesn’t matter?

This extends to anything we’re doing whether it’s launching a new initiative, working on a new set of quality criteria, or developing our architecture. It’s rarely easy to make dramatic changes in a large organization, especially when it comes to people, jobs, and rewards. In retrospect, I can look back and challenge the larger system: Were we rewarded more for making and meeting schedules or providing the most value for the company? Were people’s jobs dependent on certain products in a product year and therefore had little incentive to cancel the projects even if they were wrong? Did we have a culture that enabled us to give up a project and switch to focusing on innovation or to help out another team if it was more valuable? The answers wouldn’t have been pretty but they may have helped us start addressing the really gnarly problems in the system instead of blaming a single part of it. (In this case the product owner).

One of the things to ask ourselves when we keep focusing on one problem and it just isn’t getting better is, “what story are we telling ourselves?” What is the storyline that keeps running through our head? “If only someone would do this,” “I have to do this or else I won’t be rewarded,” “This is just the way it is.” That may be one story, but there’s always other stories to tell and we have a huge ability to change our storyline if we take a new perspective.

4. Stop Asking Permission: Empowering the Troops

To tell a more positive story, for many years I was part of a ragtag virtual team that supported each other in our madness to work in a way that made more sense to us. This virtual team had stopped really asking permission to make changes about five years ago. At that time, we saw that there were many silos of people who were doing crazy new things like Agile. It was important, we thought, to get that community talking to each other. So four of us decided to create our own conference. “Can we do that?” we asked ourselves. We supposed we could try. No one stopped us and 200 people showed up! Then we just kept doing it every year and more people showed up. Then the adoption of new practices and new culture at the company started growing exponentially. Fast forward to a year ago when we asked a senior vice president to show up to our conference and she agreed! Someone from the audience asked, “Why are there thousands of Agile adopters at the company but no one supports us?”

I’m not sure who that brave soul was, but thanks to him standing up and asking such a great question, we capitalized on the momentum and proposed a new team, again asking, “Can we do that?” Amazingly, our executives let us create our own team! We were nervous to push some of these assumed boundaries but there was no harm done and great things came out of it. We began paying it forward by helping teams to get past their own assumed boundaries and challenge the impediments. It turns out, a lot of what we do today is give people permission to challenge the things that are holding them back from being great teams.

If you've said, "this is just the way it is," ask yourself, "who said so?" If you've said, "we could never do that," ask yourself, "why not? What would it hurt to take one step forward to start changing it?" Go do something about it, even if it's a small step! You have my full permission to go try something for what it's worth. I still struggle today with my assumption of the existence of certain barriers even though I have yet to discover that any that were real. The thing is, you won't know that there's a boundary until you try to cross over it. Sometimes what's holding us back the most from making big impacts are our own preconceived ideas about what is allowed or not allowed in our companies.

5. Welcome to the Real World

"No plan survives contact with the enemy."

-Helmuth Karl Bernhard Graf von Moltke

Empowered to help groups to the best of our abilities, my team has started to explore new worlds. Specifically, we've begun to delve into the nature of our own world and the systems in which we work. Earlier this year we became engaged with learning about the Cynefin framework and the theories developed by Dave Snowden and his company Cognitive Edge. I couldn't write this paper and not introduce this framework as it has given me a completely new understanding of why Agile has worked well and what aspects of our environment it doesn't address. I won't go into great detail but I encourage you to start learning about the Cynefin framework and the nature of complex adaptive systems.

As an introduction, the Cynefin framework clearly shows the different system domains that make up our world. On one side we have ordered domains, Simple and Complicated, where there is cause and effect relationships that can be determined. In the Simple domain, we can use best practice, in the Complicated we start to use good practice and consult experts or do analysis to get the answers we seek. This is great, but it turns out there are unordered domains as well. These domains include Complex and Chaotic. Here, there is not a cause and effect relationship that can be determined. In the Complex space we can use new management techniques and tools to navigate but in the Chaotic space, we just have to act and see if anything breaks us out of the Chaos.

Simply, the Cynefin helps us to understand the nature of the system we're working in so that we can act appropriately. No domain is better than any other, it's just the way things are. However, if we assume we are in a simple system and act that way when we're really in a complex system, things can go direly wrong. Think of people gaming the draconian rules of their organization or blindly following the process without asking why we're doing the process in the first place.

Whoa! This explains so many things but demonstrates that we have so much more to learn. As I discover more about complexity thinking, I'm continually excited about the prospect that we can really effect the big hairy problems of our business that we are not addressing well today. Through this understanding, I've learned that it is most important for us to face reality. As the quote at the beginning of this section demonstrates, in both war and in development, we need to be fully aware of the situation and react accordingly. Beating a dogmatic drum is not going to make the cats go any faster in the direction you want them to move.

Since my understanding of complex adaptive systems is still a work in progress, I will defer it to a later conference. However, there are a few things I've learned based on Cynefin but also based on discussions with other coaches that I was able to apply immediately and have helped me to change my perspective in a number of ways.

5.1 Learn, Unlearn, Relearn

The first thing I learned was that I was being entirely too dogmatic, and it was to the detriment of the company. For example, our Agile pendulum had swung so far at times that managers looked like enemies and waterfall was the destroyer of all things. This was not only wrong but it pushed people away and there began to be a sense in the community of an “us vs. them” mentality. Now that I have an understanding of the different domains, I realize that there is a place for waterfall in the complicated/simple space. Agile and Lean work well in some areas of the complex/complicated spaces. And there are many more techniques we can apply depending on the system we’re working with.

I also learned about the power of human cognition and why enabling self-organizing in teams had created such large results in our organization. I realize now that we haven’t gone far enough in enabling self-organizing outside of single teams and that it’s not so scary to let go of direct control.

The third important thing I learnt was that we must stop assuming that there is a right answer to the complex problems we’re dealing with. This seems a little concerning at first, but it changes the way the problem is approached. We have to start experimenting more in a disciplined way and learn to accept that our experiments may not always produce the result we want. This isn’t bad thing, and is actually valuable. What’s important is that we are able to learn, try again, and see where it takes us.

6. Distributed Cognition

"The role of the teacher is to create the conditions for invention rather than provide ready-made knowledge."

-Seymour Papert

Managing in a complex environment is much different than managing in a simple or complicated space. While our goal is to continuously move from complex to complicated and vice versa, just wishing or pretending we can use ordered methods (hard rules, hierarchical command, etc.) will not magically transform our complex system into a complicated one. Circling back to the opening journey I went through, I realize that I was and have always been working with complex systems. Additionally, those systems became more complex as I began working with larger, interrelated groups and transformations. Had I understood the nature of systems and the methods used in complex systems, I would have saved myself a lot of heartache. It’s not simple, but at least I wouldn’t have been trying to fit a square peg into a round hole.

One of my favorite discoveries from the great work of Dave Snowden and his team is the concept of distributed cognition. It turns out that humans can figure out a lot of things if you just let them. Novel concept, I’m sure, but in practice we often completely forget this. Even as a huge champion of self-organization and empowerment, I still had many pre-conceived notions on what it meant to really let the power of human cognition thrive. Perhaps it was my ego or my lack of understanding. Or perhaps it was performance management systems that viewed only direct intervention as a rewardable behavior. Whatever it was, I had a really hard time letting go of the idea that I couldn’t be the expert on change and transformation. In fact, there is no one expert or set of experts in a complex environment! Those that think there is a right answer when you’re dealing with hundreds of people or changing the culture of an organization can think again!

In this kind of complex system, leaning on distributed cognition and the ability of diverse viewpoints to scan and better understand the system is your best bet. Now that I can accept this, I see over and over again that we are trying to put too many controls on complex systems leading them to a fragile and potentially catastrophic state. A great example of this (thanks to Cognitive Edge for bringing it to my attention), is the “Magic Roundabout” in Swindon, England (*Figure A. Magic Roundabout*). The civil engineers who developed the roundabout knew that they would need to enable movement of traffic despite unpredictable situations. The roundabout today flows traffic from a football stadium, sits right next to a fire station, and combines those facilities with five other major roads. Not only did the engineers need to enable flows of extreme variable traffic but also allow locals to get to where they needed to go quickly, protect those unfamiliar with the nature of such a roundabout from causing traffic jams and accidents, and overall ensure a very low incidence of traffic accidents.

Now, had we been using the old scientific management approach, we’d gather our traffic engineering specialists, look at the most advanced traffic light systems with proven algorithms for traffic management, and come up with a system that controls all movements of cars. After years of planning, we’d put in the fixed traffic control system all at once and stand back and appreciate our work as the population encounters it for the first time. Of course, then you would have caused traffic build up for miles as it wouldn’t be possible for such a system to anticipate and account for the high variability in traffic patterns and your government funding would have likely run out so there would be little possibility of improvements anytime soon. Additionally, there would likely be a higher incident of accidents because people are paying more attention to the rules of the stop lights than the cars around them. In fact, the Magic Roundabout solution has the added benefit of causing people to have to pay more attention to cars around them (because they aren’t relying on automated controls) and really use the power of human cognition and problem solving. One look at the final roundabout and you know it’s not something you want to go into half-asleep or talking on your cell phone. Further, the engineers didn’t just create the perfect solution from the beginning. Over a few rounds of experiments, they played with various temporary barriers and looked at the effects on the system. Once they had the system at acceptable thresholds, they locked in the barriers.



Figure A. Magic Roundabout

Applying this concept to development forced me to think, how many control mechanisms are we putting in place that actually take away from critical thinking and do not harness the power of distributed cognition? I’ve seen over and over teams that have a well-intentioned manager who simply allocates work to people versus putting the responsibility on the team to think through and practice developing robust designs. What happens is what one of my Agile coaching friends at John Deere refers to as “learned helplessness.” The team is robbed of actually learning the skills they need to stand on their own and worse, the solution design is subject to the scrutiny of only the person that developed it (the manager). Going back to the Cynefin framework, this sounds like we’ve assumed that we’re in the complicated space when in fact our solutions and the environment lie in the complex. One person can’t just come in with the “right” answer as there are many answers and we need to subject the solution to a diverse set of viewpoints.

To avoid getting ourselves into another dogmatic trap and assuming everything is complex, there are complicated environments where it is perfectly good to apply techniques that rely on expert analysis. What is important is to understand what system you're in and act appropriately given that context.

Identifying and experimenting with boundaries in the system and ensuring minimal controls while enabling the outcomes you want is a great way to do this. Then, sense what's going on with the system. What things are happening that we want to amplify? What things do we want to dampen? Continuously inspect your results and adapt from there. This notion of decentralized control with governance through boundaries is carried out more and more in the new development methods. What's critical is that we realize that the system is changing and these changes may make great improvements in our organization. But, as the organization evolves, our boundaries need to adjust to accommodate that growth.

Again, I wish someone would have told me this years ago. I regret coaching teams to just work within Scrum, as this did nothing more than enable fixed mindsets. Now we go into teams and tell them upfront that Scrum is just a method on the path to Agility and, really it is just the training wheels. They will learn and fix a lot with Scrum but eventually will pick up many new techniques as they move towards continuous flow. Ultimately, it is their responsibility to figure out why they're doing the things they're doing and to improve. They have the power! In this way, it is everyone's job to continuously improve – not just get good at Scrum and plateau after that.

7. Now Experiment!

Alright, you say, we see complexity around us but now the approach has to be one of continuous experimentation? That's a lot to ask! One major impediment we've found to developing a mindset that works well in the complex space is the fear of failure. Often companies build this fear of taking risks into the system (intentionally or not) for many reasons but, regardless of the original motivation, it exists. Knowing that we often work in a complex environment, not experimenting is actually negligent. We can no longer pretend that everything is ordered and that, if we just plan really well, we can get to the outcome we want. This idea of identifying an ideal target and then filling in the gaps simply doesn't work in a complex system. We may have a general idea of direction, but we don't need to know what the final outcome will look like as it's unpredictable. To appropriately manage in this system we need to experiment not just at the development level but at the product portfolio level and, from a change agent perspective, at our own initiative level.

This leads me to my latest mission which is not to make everyone good at Scrum or Agile or Lean but to grow a strong "Agile" mindset at our company. I'm using the term Agile as I don't have a better one (please feel free to suggest!). Ultimately, the point is that we must create an environment where everyone in the system thinks critically about the reasons why they are doing the things they're doing and, if they see that they are working in a complex domain, they experiment appropriately. Using Linda Rising's definition of Agile Mindset, based on the work of Carol Dweck, people are continuously learning and improving and accept failure as a natural part of life.

I have teams coming to me all the time simply asking for Scrum training with no understanding of why they're doing it except that they saw others be successful by using the method. I sit down with these teams and figure out what their business reasons are for wanting to change. Often, the solution may have absolutely nothing to do with the need for Scrum but rather something completely different such as helping the leadership team to define better products for their markets. I encourage everyone to ask why! By doing this, we can better avoid going through the

motions of a process (whatever it may be) without understanding the intention behind all the activities we use to get our work done. The lack of understanding of why we're operating in a certain way is a big problem I see in many teams. We have to understand what we're trying to accomplish in order to have hope of improving the way we work. Further, we need to give people permission to fail and start shifting out of a fixed mindset.

Thanks to Linda's work, I now realize that beating that drum and criticizing lack of adherence to the process was producing exactly the opposite of what I hoped to achieve, which is an Agile mindset. Now I encourage people to just start trying something new (practice) in order to change behavior and then start really experimenting. I also make it very clear whenever I'm coaching teams that this isn't necessarily easy. It will take work and some things may not work out, but it's ok because we're all learning this together. As change agents, we have to make a safe environment for people to fail and learn because, from my experience, a lot of companies typically punish this behavior. Our companies won't be successful if we don't begin to cultivate an Agile mindset. The very nature of complexity means that we don't know the right answer, so we will have to live with failures. Pretending that this is not the nature of our system is a pretty bad course that we should flee as quickly as possible.

An important thing to note is that Snowden proposes the use of Safe-to-Fail experiments. These are small experiments that won't tank your entire product or group if they fail. The purpose of these probes or experiments is to approach issues from different angles that will enable more visibility into emergent possibilities in your complex system. Start thinking of some things that you can try today to start addressing that big problem you keep running into! Within 15 minutes, I'm sure your group can come up with a couple. And don't shy away from dissent – get some people outside of your team to really hammer on the experiment and adjust it - but quickly!

Another technique that the Agile community has used for years is fast feedback loops. How can you continuously get information back that helps you know that your experiment is producing good or bad results in your system? We use pair programming, test driven development, unit level tests, system testing, customer testing, etc. as early and often as possible. As we get more into the complex space, we will have to use new techniques. But this is great! We have to know that we'll continuously be learning and adapting to the nature of the world in which we work.

8. Crisis of Faith

Don't get me wrong, after learning more about complex adaptive systems, I suddenly experienced panic. What does this mean – Agile is wrong? Have we gone too far in thinking Lean and Agile were the silver bullets? In the earlier stories of this paper, I had my identity wrapped up in being a Software Quality Engineer. Now I found myself wrapped up in the identity of an Agilist. This makes change incredibly hard as we start to question our very purpose. From years of struggling with this, I can offer only this advice: Relax!

I appreciate David Hussman's approach and mindset on this. As he says, "if it's working, use it." Agile isn't wrong, it's just applicable in different spaces particularly in transitioning complex systems to complicated. Waterfall isn't wrong, it may work very well in complicated/simple spaces. We may employ other techniques earlier in the life cycle and to enable resiliency in our company but that's perfectly fine too. For example, how does Agile help us to build our project portfolios and enable knowledge management across our company? It doesn't explicitly provide us the tools to do this! We have to use other mechanisms but that doesn't mean that Agile methods aren't fantastic in our product execution.

I'm saddened to hear of the religious wars going on in the Agile and Lean community recently. While debate is good, this type of attack on certain methodologies that are different than what we're used to isn't doing the community any favors. That's why we're seeing a lot more companies turn toward each other to learn how to take their organizations to the next level while many of the former thought leaders in the industry are arguing over old methods. There is no perfect method here. Some things work in some situations and fail in others. There are many different techniques that we should be applying based on the nature of different types of systems.

9. Conclusion

"The illiterate of the twenty-first century will not be those who cannot read and write, but those who cannot learn, unlearn, and relearn."

- Alvin Toffler

My purpose in writing this paper is to help save you from spending years of taking a dogmatic approach only to find that you aren't affecting the change you really want. I also want to let you know that if you realize you've been doing this, it's ok! You haven't failed life, you just learned from this experiment! Now take what you've learned and try something different; see what happens! The biggest risk we run is getting stuck in a fixed mindset. It turns out that if we're working in a complex domain, we won't have the answer. This is a very freeing concept if you think about it. One great step to take is to begin to understand the nature of systems and techniques for working in the complex domain (as this is one of the more unfamiliar domains). You can also cultivate an Agile Mindset and help those around you to use an Agile Mindset as they are approaching their day to day work. Good luck and remember to ask why, experiment, and learn!

References

Avery, Christopher. "Responsibility Process." <http://www.christopheravery.com/responsibility-process#> (accessed August 11, 2013).

Cognitive Edge. <http://cognitive-edge.com/>

Snowden, David J. and Mary E. Boone. 2007. "A Leader's Framework for Decision Making." Harvard Business Review (November).

Rising, Linda. "The Agile Mindset – What's Next?" 2013. https://submissions.agilealliance.org/system/sessions/attachments/000/000/221/original/PNSQC_Mindset_-_60_min.pdf. (presented at the Agile 2013 Conference).

"The Magic Roundabout." <http://www.cbrd.co.uk/indepth/magicroundabout/> (accessed August 11, 2013).

"Safe-to-Fail Probes." Cognitive Edge. <http://cognitive-edge.com/library/methods/safe-to-fail-probes/>. (accessed August 21, 2013).