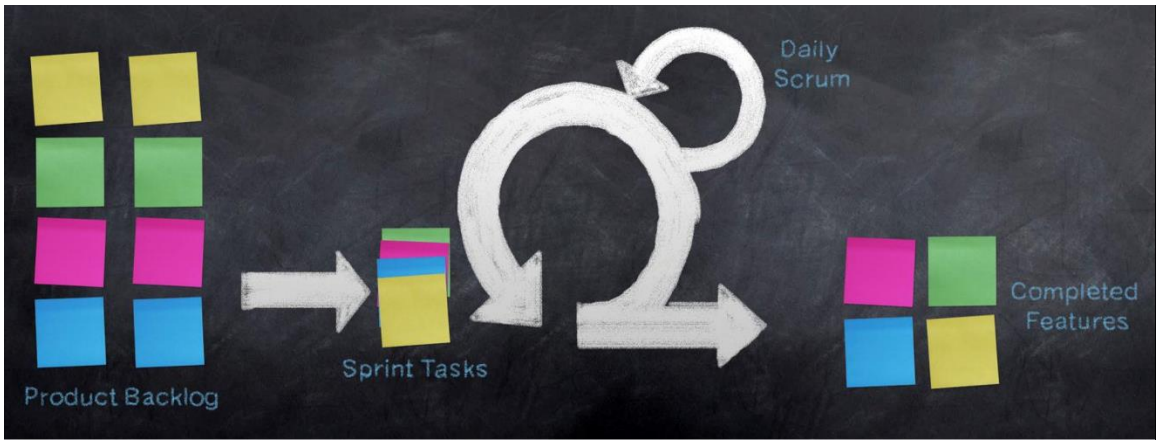


Delivering Value

The Benefits of Cultural Agility

Robert Weidner

Table of Contents

ABSTRACT

With the rapid pace of change and technological advancement, many companies are searching for a way to do more, with less, faster than ever before. This is driving the need for an adaptive versus a predictive model that can quickly respond to changing needs and emerging market conditions. This paper will explain how companies can begin to implement cultural agility to help them deliver more value to their customers. We'll examine what it takes to transform, the principles and practices that have helped other organizations achieve their goals, the large body of comparative research that has been done on Waterfall and Agile methods, the leadership orientations that have proven successful in this new paradigm, and how the organization can gain maximum efficiency out of their team structures and space design.

INTRODUCTION

Let me begin by telling you a little about myself, as I'd like you to understand at the outset that these are not the words of a mere theoretician. What I am about to share with you has *practical application*.

My name is Robert Weidner (Wide-ner). I began my career as an enlisted member of the armed forces serving in the United States Navy, with a Top Secret SCI security clearance and a rate designation of Information Systems Technician. I was a plankowner of the USS Harry S. Truman, an aircraft carrier that was – at the time – the largest ship in the world. After an honorable discharge (and a service connected disability), I began my career in the civilian sector.

While raising a family and working full time, I also took college classes in the evening. At first I found employment as a Network Administrator, then a Systems Engineer, and eventually became a clinical Project Manager. As a strong believer in Aristotle's notions of 'civic virtue', I decided to focus my career on the healthcare sector, where it felt like I was truly delivering something of value back to the community.

As a Project Manager I built my reputation by delivering against the most difficult, complex projects. Many of these efforts were handed off to me as others failed to complete them. I remember one project in particular that had been languishing for nearly a decade, and ate through six other Project Managers before landing on my lap. Working closely with business stakeholders, we were able to break through the longstanding stalemate to reach consensus on difficult contract terms, and delivered a new, fully integrated chemotherapy order writing system across 10 different sites in a compressed timetable.

Other significant efforts that I have been charged with included a redesign and build of all clinical applications for an electronic health record, the implementation of claims-based authentication, synchronization services, automated user provisioning, a development platform technology refresh, the creation of a baseline SDK for all new clinical applications, numerous application conversion and recompiling efforts, the implementation of Lab systems, insurance payer systems, and a myriad of other solutions.

In short, I understand effective project management. I'm also all too aware of the pitfalls that often accompany it. No matter how good I was at my job, it seemed as though scope, schedule, and cost changes were inevitable. So much of what happened was outside my – or even the project teams – ability to control.

Then I was introduced to agile. Suddenly, the collective failure in our thinking became clear to me. Our attempts to manage these changing conditions through stricter controls and a more detailed plan were erroneous. We weren't just *doing* it wrong, we were *thinking* about it all wrong. I dove head first into this new world, where everything was based on delivering direct – rather than indirect – value (more on that later). It became a passion for me… my life calling, so to speak. I learned as much as I could, as fast as I could. Furthermore, I benefited from having some great Agile mentors along the way. Since then,

I've led 14 different Agile transformations for various organizations and affiliates, predominantly within the healthcare sector.

That's what this paper is about. Not just doing Agile. Not just thinking Agile. But *becoming* Agile.

Coaching Biography

Recognized subject-matter-expert in agile software development:

- 20+ years of experience in Information Technology
- Has applied agile and lean thinking to multiple industries, from startups to Fortune 100 companies
- Trained thousands of people and coached hundreds of teams
- Whitepapers published on topics ranging from Agile Adoption to Open Space Design
- 10+ industry related certifications including SPC 4, PMI-ACP, CSM, CSP, and PSM III
- Became the 28th Certified Enterprise Coach (CEC) in the United States
- Frequent guest speaker at user groups, conferences, and universities
- Adjunct instructor for graduate and undergraduate courses in Organizational Effectiveness, Managing Project Teams, and Technology Strategies for Healthcare Management
- 7 college degrees, including: Master of Science in Organizational Change Leadership (MSOCL), Master of Arts in Counseling (MAC), Bachelor of Arts in Business Administration (BBA), Bachelor of Arts in Computer Science (BACS), Bachelor of Arts in Marketing (BAM)

Lead Enterprise Coach for 14 agile transformations:

- Optum Technology
- National Marrow Donor Program – Be The Match
- Marshfield Clinic
- MCIS
- Security Health Plan
- Biomedical Informatics Research Center
- Lakeview Medical Center

TRANSFORMATION

Agile transformations are surprisingly difficult. According to Harvard Business School professor and author, Dr. John Kotter, only .001% of organizations have actually succeeded at becoming culturally agile. But he's also quick to explain:

> *"It doesn't have to be that way! It doesn't. You can change. You can create what's needed. And when is the time to do that? No question… it's now!"*

To get started on this journey, it requires courage. The courage to step out of your comfort zone and learn to do things in a different way. The courage to experiment, and shift course when something isn't working. The courage to grit your teeth and push forward when the going gets tough. Change is hard work, and it doesn't happen of its own accord. You must will it. As George Bernard Shaw once wrote:

> *"The reasonable man adapts himself to the world; the unreasonable one persists in trying to adapt the world to himself. Therefore, all progress depends on the unreasonable man."*

The Courage to Change

When I was in eighth grade, the library in the small town I grew up in hosted a paperback giveaway. I filled a grocery sack with all the books that looked even mildly interesting. Over the next few months I began to devour this collection, which included a small novel by Richard Bach entitled, Jonathan Livingston Seagull. At first I thought the book might be non-fiction, as it contained numerous photographs of seagulls in flight, blurred by their rush of movement. Instead, it was a simple and straightforward but deeply compelling story about a seagull who sought, above all else, a purpose-driven life.

Rather than scrounge, fight, and squawk with the other birds for food, Jonathan wanted to learn how to fly. He didn't just want to flap his wings and move from point A to B, but rather to embody perfection. In his effort to do so, this odd bird became outcast, viewed by the elder as reckless and dangerous. What he wished for the flock he now gained for himself, discovering new and expansive methods for flight. These techniques allowed him to retrieve fish from deep below the surface of the water, or to dine on insects from weather patterns that the others never even knew existed. Jonathan no longer had to depend on scraps thrown to him by fisherman. He could live an independent, self-directed life, focused on the pursuit of perfection.

As his eventual, like-minded friend, Sullivan, would remark: "The gull who flies highest, sees farthest." Jonathan was the gull who flew highest, learning in one lifetime what took others a thousand lifetimes to learn, and by doing so, he was the gull who could see farthest. But he couldn't help but think that one or two gulls from his old flock, with their feet planted firmly on the ground, might feel the way he did. What if they, too, sought perfection above all else? How much more would he have known if a teacher had come to him in those early days? How much higher could he have climbed? How much farther would he have seen?

These questions continue to nag at Jonathan until one day he decides to return to his old flock, in the hope that he can impart some of the knowledge he has gained to a young gull who might be searching for a deeper meaning to their life. It is here that he finds a slew of birds, all of them made outcast, all of them searching for purpose, but none more sore than Fletcher. Through Jonathan, Fletcher comes to learn the most important lesson of all: love. Even for those who do not love you. Even for those who made you outcast. For they cannot see that they only hurt themselves. He must forgive them. He must try to teach them.

Jonathan, by the way, is the name I gave to my son when he was born, in the hope that he would also one day find his passion and that this might bring deep meaning to his life. This should illuminate how much this story has influenced my own perspective and shaped my worldview. I love not to be loved in return, but for the sake of love itself. I show love by teaching, trying to impart any knowledge I have to others in the belief that it may further enrich their lives. The meaning of my life can be summarized succinctly by the following two words: *continual growth*. I seek to become more than I am, and have a strong desire to help others do the same.

I have often had trouble sleeping. My mind is a whir of gears that are constantly turning, as I revisit the events of the day, or think of new and exciting ideas. I have since studied meditation to learn how to quiet my mind, to gain discipline over my thoughts. Not that it always works. But before that, I would read myself to sleep in order to divert my thoughts to some known topic of consideration. Rather than have my mind buzz of its own accord, I would read or listen to an audiobook that I was well acquainted with, and repeat each line as though it were a mantra. I have come to memorize many great passages this way, but none more so than Jonathan Livingston Seagull, which I must have read over eighty times. It was the perfect solace for a restless night, as I could complete the entire thing in under thirty minutes, and for all its simplicity, it contained within it deep layers of meaning.

The lessons from this remarkable book remain with me still. Later, I would draw further inspiration from other works by Richard Bach including, Illusions: The Adventures of a Reluctant Messiah, and Running from Safety. But it was a small, anecdotal story written in his journal and later published under the title A Gift of Wings that I would often think of in the years to come.

It was in this autobiographical book that Richard would tell about a homeless man he had met in the streets of New York City. The man had no family, no house, and no career. There was nothing that tied him to the area. Why, asked the author, do you stay here, with the harsh winters and the lack of hope? Why not go where it is warm, or where it is beautiful? Why not visit the beaches of Florida, or the mountains of Colorado? Why not walk where you can feel the grass beneath your feet?

Because, came the man's paraphrased reply, *I lack the courage… I fear what I don't know more than I despise what I have*.

Since first reading those words, I have repeatedly witnessed this phenomenon. While earning a Master's degree in Counseling, and then a second one in Organizational Change Leadership, I have frequently read in textbooks and been told by instructors that people fear change. It is, they say, human nature. In my line of work, I am often confronted with this reality.

You see, I am a change agent. I transform organizations; by helping them focus on human interactions, process efficiency, and continuous improvement, in order to develop better products and deliver better services. In short, I am an Enterprise Agile Coach, and I teach cultural agility. In doing so, I often meet those who are reluctant to change. These are people who have worked a certain way for many years and have become very good at what they do. Suddenly, I present them with a very different model and it quickly becomes clear that they are no longer the expert, but the student. The immediate response is often one of apprehension. Even when the reasons for change are quite clear. *We fear what we do not know more than we despise what we have.*

This is an over generalization, of course. While it is true that many people fear the unknown, it is also true that there are those who embrace it. Those who seek to expand their horizons and learn as much as they can during their short time on this earth. I am of the latter mold, and while seemingly in the minority, I occasionally meet others who are cut from this same cloth. We are more like the character of Jonathan, seeking perfection through continual growth and adaptation. Stagnation, to us, is death; complacency the death knell.

Businesses that wish to survive would do well to avoid complacency. They should strive for continuous improvement through perpetual adaptation. This is achieved, as Jonathan showed us, through the pursuit of perfection. To me, this is – quite simply – a better way to be. But to get there, we must find the courage to change.

Which Shadow Do You Cast?

The same mantras are repeated in businesses across the globe: we must do more with less; we must go faster. But what if I told you there was a way to work smarter, instead of harder. Would you be open to making this change?

There is a smarter product development framework, and many organizations have already discovered it. However, for the large number of companies who have implemented this framework, there are few who actually do it well. In our learning economy, this is the difference between those that survive, and those that succumb to some emerging startup with the raw skills and innovation to capture their share of the market. Or all of the market, for that matter. Look out auto America, I would like to introduce you to the Toyota Production System. Move over Alta Vista, here comes Google. Get out of the way Microsoft, make room for Apple.

For the first time in history, economies of scale are actually working against corporations as they struggle to adapt fast enough to the ever-increasing rate of change that occurs in our modern society. The rate of technological discovery increases at an exponential—

rather than linear—rate. (Kurzweil, 2005). The larger the organization—the more structure it has, the longer the chain of command, the added red tape—the more difficult it will be to quickly adapt to emerging market conditions. In an economy based on knowledge work, innovation becomes the currency of the day, and those organizations with obstructive change management processes and economies of scale will be left wondering how David got the best of Goliath. The answer is simple: David moved faster. The question then becomes: how can we, too, move faster? How do we prevent our size and rigidity from becoming our death knell?

The answer is distributed processing; by which I mean empowerment throughout the lowest levels of the organization. If there is anything a well-educated and highly skilled employee knows, it is the details of their job. Stop telling them *how* to do it, and start focusing instead on *what*. Let your customers decide your product features, by gathering empirical and incremental feedback throughout the development cycle, and leveraging adaptive planning techniques to create a product that truly meets their needs. Even if those needs should change or evolve over time.

Unfortunately, implementing a few practices will not be enough to transform your organization. To ensure success and sustain the change, you must achieve a principle-based implementation. If you focus solely on practices, the disconnect will show through, and your organization will be in a state of friction. After all, no one respects a hypocrite. The goal is not about *doing* Agile, but about *becoming* Agile. This is the difference between *Agile in name only* and *Agile done well*. If your organization focuses only on practices, then they will come to represent the former. Meanwhile, if they focus instead on principles that align to those practices, they will achieve the latter. It is as simple as that. Which of these two organizations do you reside in? If it is the former, in what ways can you influence it? Which shadow do you cast?

Simple is the Hardest Thing to Do

A few years ago I started a consulting company that focuses on helping organizations achieve cultural agility. The mission statement is simple: "We are dedicated to continual improvement through perpetual adaption. Our mission is to help others evolve." While Agile may seem simple on the surface, it is extremely difficult to do well. Our friends, family, and co-workers will often remind us to "keep it simple", but in my experience, simple is the hardest thing to do. It's in our nature to overcomplicate matters, because it gives us the illusion of control.

Agile, which often seems simple on the surface, is one of the most difficult transformations to undergo. In most organizations, adding process steps is relatively easy, compared to removing them. Once a new procedure goes on the books, how quickly are you able to remove it later? How many committees must you call? How many signatures do you need? We are trapped by time's arrow, in which it's relatively easy to move forward, but seemingly impossible to move backward. (Greene, 2004). The goal of an Agile transformation is not to scale an organization, but to descale one. (LeSS, 2015). This is like travelling backwards in time to remove processes, procedures, and methods that have been developed over the course of decades. The goal is to simplify the organizational structures,

lean out the processes, collapse feedback loops, and ensure value delivery. This is no easy task.

I often hear the same reasons given by leaders at companies who wish to transition to Agile: "We'd like to reduce our cycle time, increase our speed to market, and simplify our environment so we can more readily adapt to changing market conditions and customer demands." However, these leaders frequently mistake agile for a process, rather than a set of principles. By doing so, they shine the light of transformation outward, but not inward. They ask others, "What are you doing to become more Agile?" but fail to scrutinize how their own behaviors and actions support or hinder the overall change effort. This can set the organization down the wrong path.

There is often a mistaken belief that an Agile transformation is about replacing the legacy Waterfall methodology with an Agile framework, and the impact will be limited solely to the software development phase of the overall product life cycle. The truth is, process is the smallest and easiest component of an Agile transformation. What we are really changing is the *culture*, and this is what will determine if the transformation succeeds or fails. Cultural agility is about a shift in mindset, and this shift has far-reaching implications for the company. It should fundamentally change everything about the way products are developed, from the funding model to hiring practices and reporting structures. It is, by definition, *transformational* change.

The agile mindset is one of the most significant paradigm shifts an organization can make. Lean and agile principles must become embedded within the culture itself. Each individual in the organization will undergo his or her own micro change effort, which will either contribute or hinder the overall macro effort. Therefore, communication becomes key. It is imperative that we focus on both the macro and micro perspectives, and help people move from exploration, to insight, and then into action. In a large organization, this shift can often take several years.

We are at the trailhead of what can often be a difficult and arduous journey, but which is also infinitely rewarding. The outcome will depend on our ability to transform not just the organizational methods that are employed, but in our ability to shift how each member thinks and behaves. There is a substantial difference between *doing* Agile and *being* agile. We must effectively communicate the value of agile to keep people engaged, and inspire them to take this journey with us.

What Did I Sign Up For?!

Mike Cottmeyer, CEO and founder of LeadingAgile, said in a recent interview that executives often don't realize what they're signing up for when they make the decision to 'go Agile'. Mike explains:

> *"Here's what I think's happening. Agile gets sold as a process. So if they teach you the process you'll get the benefit of the process. The reality is the process only works in the presence of the ability to form teams, create backlogs, and produce working, tested software. Executives are signing up*

> *for Agile because they want the benefit, and they've been told that Agile will give them these benefits. They didn't sign up for a total refactoring of their enterprise into complete cross-functional teams and everything that goes along with that. They didn't sign up for the level of engagement necessary and the program and portfolio stuff necessary to create balanced portfolio backlogs. They didn't sign up for the amount of refactoring and technical validation to be able to do working and tested software. They signed up for Scrum training. So the challenge for us as a community is that we have to be more direct with the executives we are engaging and go look, you've signed up to do agile, right? We can teach this organization agile, but let me tell you 80 or 90% of the impediments we know you're going to hit so when we hit them, [you have also] signed up to fix them. They've signed up for going to agile, but they haven't signed up for reducing the impediments. We need to be honest with them about the challenges they're going to see as they go down this path."*

Mike claims that the key to becoming agile is to reduce the number of dependencies between teams:

> *"The problem is, some of these dependencies are organizational. They can be fixed. Some of them are technological. So if you think about the architectures and the platform… let's say you're a COBOL team working on this 40-year-old platform, and you've got millions and millions of lines of code, and there's 18 teams that are working with it, and any one team could break something that impacts you – there's no good testing, there's no continuous deployment – and so… how agile can you be? You can do Scrum, but there's no agility there."*

Mike puts forth the case that only three things are needed to transform any size organization to Agile:

1. **Form teams**
2. **Create backlogs**
3. **Produce working, tested software**

According to Mike, the problem is that organizations often attempt to apply Agile to their current team structures, which are not cross-functional:

> *"A team is a group of people that has everything and everyone necessary to produce a working, tested increment of software. It's an encapsulated value stream. They can take it from idea all the way to deployment. And that's the magic of it, because when you have a team that's formed that way they don't have any dependencies with anything in the rest of the organization. They can operate with autonomy. They can inspect and adapt. They can do all these different type things. And so when you don't have teams that are formed that way – they're matrixed, or they're part time, or they're missing key leadership, or they don't have autonomy over the technical stack, or*

> *any of the number of impediments that get in the way of forming teams – then you don't do Agile. Ken Schwaber, at one point in time, said something to the effect of Scrum is like chess. You don't argue whether Scrum works… you either play it well or you don't. What we're trying to do is play Scrum without having the right pieces in place, and the playing board of Scrum is a well formed team. Almost every failure mode in Scrum tends to start with a team that is not complete and well formed. It's just that simple. Large organizations trying to adopt Agile are absolutely failing to do this over and over and over. A transformation journey is about figuring out how you want to form teams, forming them, and then progressively starting to decouple them from each other over time. That's the only way."*

I firmly believe the path to a successful Agile transformation is to eliminate handoffs and phase-gates through the creation of static, cross-functional teams that are feature-based, and then to collapse the feedback loops as much as possible. This often means moving from a project to a product model. To do this effectively, it requires a shift from functional silos and component-based teams to cross-functional structures. This doesn't have to be done all at once, but you do need to start somewhere. Furthermore, if you really want to reduce cycle time and increase speed to market, then you need to look at the entire product lifecycle through a Lean-Agile lens, from ideation through delivery and even beyond, all the way into support and maintenance. Reducing cycle time only within the development phase results in a very minor reduction overall, particularly if you still have an elongated budgeting cycle from ideation to funding approval, or a lengthy release process that follows the development effort. You need to map the entire value stream, and remove waste — defined as anything that does not add direct value to the customer — from throughout the *entire* system. This — combined with a shift towards iterative and incremental development that requires participation and collaboration with the customer — will fundamentally change *everything* about the way in which your company develops products. This doesn't just impact a phase in the product lifecycle… Agile *replaces* the product lifecycle. Thus, if you're company is not taking an enterprise view, they are significantly hindering their chances for success.

Every upstream and downstream process (from funding to release management) within your organization has likely been designed — over the course of many years — around a traditional, Waterfall methodology. The processes are often sequential, and exist as phase-gates that produce significant friction when transitioning to Agile. Until those upstream and downstream processes can be brought into alignment, teams will continue to struggle. If this lack of alignment persists long enough, most organizations will revert back to their previous state of existence and exclaim, "We tried Agile, and it didn't work!" Often, they fail to realize that they tried *Agile in name only*, having changed none of the upstream or downstream processes, or the culture itself for that matter. When Agile succeeds, it does so because the organization took a holistic view and approach to the implementation, and aligned principles to practices. It requires a simultaneous top-down and bottom-up transformation strategy. This creates alignment and provides a conducive environment for the change to take hold. Without doing so, the implementation will be destined to fail.

The Agile Catch-22

Many organizations are structured by either function or domain. Often, if they exist in the software development industry, their applications – which do not have a one-to-one correlation to a product – have component-based application teams that support them. Thus, if leadership focuses on the application level and attempts to transform the corresponding teams from Waterfall to Agile without doing any type of organizational restructuring, then they haven't created cross-functional, feature-based teams. Consequently, no handoffs will have been eliminated, and the feedback loops will not have been shortened (save for perhaps one, if the team has gained an embedded Product Owner). Thus, their ability to reduce cycle time and increase speed to market will have been severely hampered, the teams will be unlikely to show a significant difference in productivity, and the organization is therefore less likely to gain the buy-in needed to expand the overall change effort. Development teams will inevitably become frustrated as they encounter cultural and process resistance. At this point, all that's been done was to teach these teams a few practices… but those practices are no longer in alignment with the principles that the organization runs by. The "process" (due to the high level of transparency it creates) will start to get blamed for the resulting churn, oppositional voices will grow louder, and in time leadership will decide to do one of the following to overcome the impasse:

- Abandon the "Agile experiment" all together *or*
- Move towards cross-functional, feature-based teams

My recommendation is to *start* with the latter, and avoid the mistakes that so many other organizations make when they undergo an Agile transformation.

To further compound the issue, most organizations use a project-based funding model that takes a *program* view of the work. Think about this for a minute… they have an organizational structure based on *domains*, teams aligned with *applications*, and funding based on *programs*. What does this layered complexity mean for an Agile transformation? Let's break it down:

- Teams are predominantly component-based, aligned to an application or application area, and they report up through a domain
- Funding is allocated to programs, which may or may not cut across multiple domains, as well as applications (and therefore application teams)
- Scheduling and Capacity/Demand Management are allocated around a project-based model, rather than static teams with a singular backlog

In the common scenario described above, the program leverages, but does not own, the application teams. Thus, while the funding is approved for the program, they have little control over moving towards static, cross-functional teams. Instead, their work simply goes into multiple queues for each application team that is required to deliver the necessary feature set. The work is scheduled via capacity and demand management, based on existing team structures.

Meanwhile, if the change agents focus on the applications and teams themselves, they run into the following issues:

- They would likely increase the complexity of those in-flight, multi-application

programs (virtually ensuring that they would need to use some sort of mixed methodology model)

- They would be taking a thin-slice view of the overall transformation effort, which would likely result in "blind spots”
- Many of the applications wouldn’t be significant enough to qualify as “Products”, and therefore would not warrant a dedicated team with a Product Owner
- Getting to a cross-functional team would still be difficult, as it requires that they look across multiple team structures and skill sets

While the best option is always to restructure around products, forming cross-functional teams, not all organizations are willing to take this “drastic step”. I often hear leaders proclaim they are transforming their organizations to Agile, but then they fail to make the bold moves that are necessary for transformation to truly occur. While they talk about transformational change, what they do is merely incremental change. In most cases, this is a slow road to change, and a fast road to failure.

So… if Agile isn’t a process, then what is it?

AGILE

Seventeen pioneers in the information technology industry gathered at a retreat in Snowbird, Utah, on February 13, 2001; where they drafted and signed the Agile Software Development Manifesto. The group was determined to codify the unifying principles that each of their frameworks adhered to. The documents signers included Kent Beck, Mide Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, and Dave Thomas.

Each of the authors represented various lightweight frameworks that used an iterative and incremental approach towards product development in order to validate hypothesis. These frameworks included: Extreme Programming (XP), Dynamic Systems Development Method (DSDM), Adaptive Software Development (ASD), Crystal, Feature-Driven Development (FDD), Scrum, and Pragmatic Programming. Other frameworks and practices have since emerged that subscribe to the same set of values and principles. These values and principles are:

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Principles Behind the Agile Manifesto

We follow these principles:

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity – the art of maximizing the amount of work not done – is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective and then tunes and adjusts its behavior accordingly.

LEAN THINKING

Agile frameworks have emerged largely as a derivative of Lean thinking, where manufacturing techniques for plant optimization have influenced product development within the software industry. Lean, which originated as a set of tools and techniques, has since evolved into a set of philosophical principles. The term was coined by Womack, who used it to describe a system that does more with less. Lean is a customer-focused process that emphasizes value-delivery and eliminates any waste that does not directly support this aim. Womack and Jones defined the five principles of this technique, which later were adopted as part of the Toyota Production System (TPS). TPS, in turn, served as an inspirational model for Scrum (along with the PDCA cycle and OODA loop), when Hirotaka Takeuchi and Ikujiro Nonaka published a paper in 1986 in the Harvard Business Review entitled, The New New Product Development Game, which described this emerging Japanese business culture that used an overlapping process instead of the NASA-invented Waterfall methodology. This paper provided Scrum with its namesake, as the authors compared the overlapping process to the work of a rugby team, stating that the best teams act as though they are in a scrum: "...the ball gets passed within the team as it moves as a unit up the field."

The Five Principles of Lean

Lean thinking involves specifying value, lining up activities in the most effective sequence, conducting activities without interruption whenever 'pulled' to do so by customer demand, and continuously seeking ways to improve.

- **Value**
  - This needs to be viewed and defined from the perspective of the end customer.
- **Stream**
  - Next, the value stream needs to be mapped and wasteful steps eliminated.
- **Flow**
  - For the value-creating steps that remain, the team then seeks to optimize the flow.
- **Pull**
  - Flow is pulled through the system to deliver what the customer wants, when they want it, without stockpiling waste.
- **Perfection**
  - Perfection is the elimination of waste. The team seeks continuous improvement in pursuit of this goal.

The Seven Wastes

Taiichi Ohno's seven wastes offer employees a useful framework for searching out and eliminating activates that fail to add value.

- **Overproduction**
  - Making more than required, or making it sooner than required.
- **Waiting**

    - Products waiting on the next production step, or people waiting for work to do.
- **Unnecessary transportation**
    - Moving products longer distances than required.
- **Over-processing products or parts**
    - This can occur because of poor design or inefficient tools.
- **Inventory**
    - Holding more inventory than is required.
- **Unnecessary motion**
    - People moving or walking more than is required.
- **Defective parts**
    - Requiring effort to inspect and fix.
- **8th Source of Waste**
    - Producing goods and services that fail to meet the needs of the customer.

Kanban

Kanban is a pull-system that uses a "signal board" to visually display the status and priority of work in process. Characteristics of Kanban include:

- **Ability to visualize the workflow**
    - This is an advantage, since what cannot be seen cannot be improved. With Kanban, you can show progress in real time.
- **Kanban allows leadership from a team approach**
    - This is based on the idea that, without collaboration, nothing of significant value can be created or improved.
- **A focus on reducing the batch size**
    - Decreasing the size of the batch unit of work increases the flow through the system, particularly at the constraint.
- **The opportunity to learn and continuously improve**
    - Teams can take time to reflect in order to learn from experience.

The Seven Principles of Lean Software Development (LSD)

Following are the principles of Lean expounded upon and applied to software development and frequently referenced within the Agile community.

- **Eliminate waste**
    - Anything that does not add value to the customer should be regarded as waste and subsequently eliminated.
- **Amplify learning**
    - Speed up the learning process by using short iteration cycles to collapse the feedback loop and amplify learning.
- **Decide as late as possible**
    - Defer commitment for as long as possible, making decisions when the greatest amount of information is available.
- **Deliver as fast as possible**
    - Identify and solve business needs as quickly as possible, then iterate on the design based on feedback.

- **Empower the team**
  - Those close to the technology on a daily basis are in the best position to make technical decisions. Allow them to do so.
- **Build integrity in**
  - Involve the customer throughout the development effort to validate assumptions and ensure quality.
- **See the whole**
  - Produce a system with smoothly interacting components through continuous integration and by removing technical debt.

Lean thinking provides a contextual frame of reference for how we should view the activities we perform as part of a series of development steps. If these steps do not add direct value to the customer, then perhaps they should be eliminated. Otherwise, they will continue to overcomplicate the system and slow down production. Using the business philosophy of kaizen, the goal is to continuously improve the system as a whole, through small feedback loop adaptations. In Lean, it's common to defer commitment until the last responsible moment. We should never make decisions early, when we know the least, as this will box us into a corner and decrease our ability to be agile. Early decisions based on a lack of empirical knowledge merely limit our options down the road. Furthermore, documentation should be barely sufficient. Which is to say sufficient… but barely. Focus on direct, not indirect, value.

This is what it means to think Lean. Our decision-making processes should be derived and heavily influenced from this perspective. This is one of the hardest changes for people to make, particularly those who are used to over-producing. We make decisions early, because we believe that's the sign of a good leader. We ask teams to commit while underestimating the complexities of their work. We require that they document everything as a CYA measure prior to handing if off. We falsely believe that our value to the company exists in these indirect value-add steps, rather than in delivering what provides direct value to the customer. Thus, not only do we have to *do* things differently, we also have to *think* about them differently. The more our thinking begins to influence our actions, the more this shift in mindset will become a core part of our being. Soon, it will simply be who we are. But at the start of our journey, it is precisely who we are not.

THE SCRUM FRAMEWORK

Agile is a philosophy, like existentialism. Meanwhile, Scrum and other frameworks are a method for utilizing those philosophical principles, and then applying them in practice. Where Lean thinking helped influence some of the philosophical underpinnings for Agile software development, Scrum became the way many organizations chose to align their practices to these newly adopted principles. Scrum's emphasis on the creation of a collapsed feedback loop cycle is ideally suited for a complex work domain, where there is a need to shift from predictive planning techniques to adaptive planning in order to create a competitive advantage.

According to the annual State of Agile survey, Scrum is the most popular and widely implemented Agile framework, with 68% of respondents using it within their organizations (Version One, 2016). Jeff Sutherland and Ken Schwaber, the co-creators of Scrum and authors of The Scrum Guide, define Scrum as (2016):

Definition of Scrum

Scrum (n): A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value.

Scrum Theory

Scrum is founded on empiricism, which asserts that knowledge comes from experience and making decisions based on what is known. Three pillars uphold every implementation of empirical process control: transparency, inspection, and adaptation.

- **Transparency**
  - Significant aspects of the process must be visible to those responsible for the outcome.
- **Inspection**
  - Users must frequently inspect artifacts and progress toward a Sprint Goal to detect undesirable variances, at the point of work.
- **Adaptation**
  - If an inspector determines one or more aspects of a process deviate outside acceptable limits, then the process must be adjusted.

Scrum Values

Successful use of Scrum depends on people becoming more proficient in living these five values.

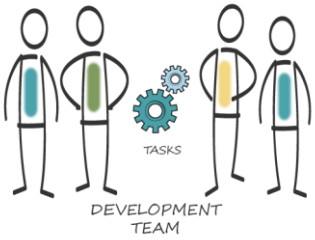
- **Commitment**
- **Courage**
- **Focus**
- **Openness**
- **Respect**

The Scrum Team

Scrum Teams are self-organizing and cross-functional. They deliver products iteratively and incrementally, maximizing opportunities for feedback.

- **Product Owner**
  - Responsible for maximizing the value of the product and the work of the Development Team.

- **Development Team**
  - Professionals who do the work of delivering a potentially releasable increment of "Done" product at the end of each Sprint.



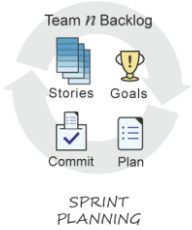
- **Scrum Master**
  - Responsible for ensuring Scrum is understood and enacted.

Events

Prescribed events are used to create regularity and minimize the need for meetings not defined in Scrum. All events are time-boxed. Each event is a formal opportunity to inspect and adapt something.

- **The Sprint**
  - The heart of Scrum; a time-box one month or less during which a useable and potentially releasable product increment is created.

- **Sprint Planning**
  - Answers the questions: "What can be delivered?" and "How will the chosen work get done?" for the upcoming Sprint.



- **Daily Scrum**
  - A 15-minute time-boxed event for the Development Team to synchronize activities and create a plan for the next 24 hours.

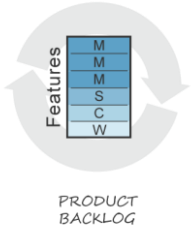
- **Sprint Review**
  - Held at the end of the Sprint to inspect the Increment.

- **Sprint Retrospective**
  - An opportunity for the Scrum Team to inspect itself and create a plan for improvements to be enacted during the next Sprint.

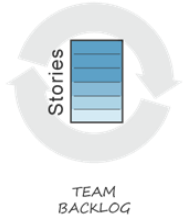
Artifacts

Represent work or value to provide transparency and opportunities for inspection and adaptation.

- **Product Backlog**
  - An ordered list of everything that might be needed in the product, the single source of requirements for any product changes.



- **Sprint Backlog**
  - The set of Product Backlog items selected for the Sprint, plus a plan for delivering the product Increment to realize the Sprint Goal.



- **Increment**
  - The sum of all the Product Backlog items completed during a Sprint and the value of the Increments of all previous Sprints.

Backlog Details

Items are moved from the Product Backlog to the Sprint Backlog during Sprint Planning, based on priority, velocity and capacity.

- **Sprint Goal**
  - An objective set for the Sprint that can be met through the implementation of Product Backlog.
- **Product Backlog Item (PBI)**
  - A unit of work or business value (i.e., product requirement) contained on the Product Backlog.

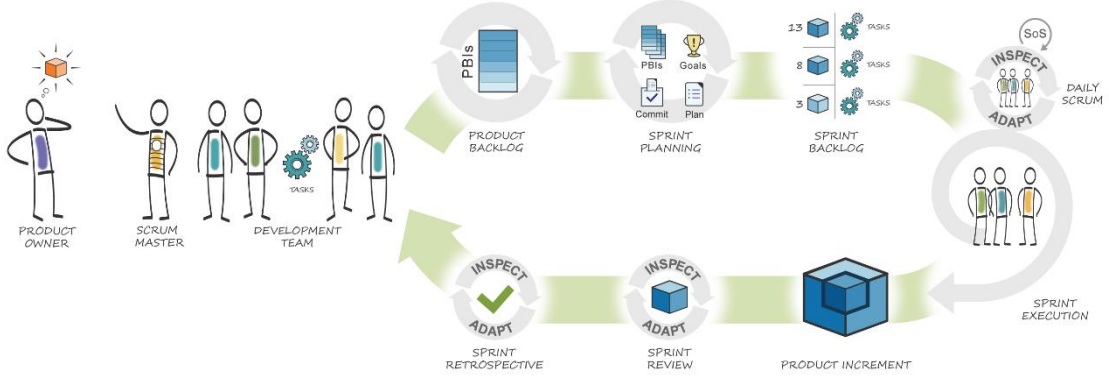
Artifact Transparency

Decisions to optimize value and control risk are made based on the perceived state of the artifacts. To the extent that transparency is complete, these decisions have a sound basis.

- **Definition of "Ready" (DoR)**
  - Product Backlog Items that can be "Done" by the Development Team within one Sprint are deemed "Ready" for selection in Sprint Planning.
- **Definition of "Done" (DoD)**
  - A shared understanding of what it means for work to be complete.

End Note

Scrum is lightweight and simple to understand, but difficult to master.



WHY AGILE

You may be wondering: in what ways does Agile differ from the traditional, phase-based and sequential approach of Waterfall? Why would an organization want to use adaptive planning techniques rather than predictive ones?

The truth is – if the work you're doing is simple and can be planned with total accuracy up front – then you wouldn't need Agile. However, most of the work we do is complex, and increasingly so. We are inundated with statistics of high-cost Waterfall projects that – even if they are on time and on budget – still fail to deliver the promised value back to the organization. Scope is planned in advance, obstructive and costly change processes are put in place, and in the end something is delivered that may have been needed when it was requested, but is no longer relevant based on current market conditions and business realities.

How is Agile Different?

Agile frameworks provide a toolset to help manage change, and deal with ambiguity. It makes transparent and explicit the things that other methodologies only imply:

- Projects do not follow a straight line from start to finish.
- Projects are too complex to define all business requirements up front.
- Requirements are likely to change.
- Time estimates are really guesses, and should not be contractually binding.
- The customer needs to be involved and the team needs to be willing to shift direction when necessary to deliver a product the customer actually wants.
- Emergent design (i.e., bottom-up), is more efficient, effective, and innovative than top-down designs.
- Prioritization of features and collaboration ensures the development team is always delivering the most value back to the business.

In contrast to Waterfall, Agile frameworks:

- Manage complexity through iterative and incremental design.
- Minimize rework and adapt to changing business requirements using just-in-time detailed analysis.
- Introduce flexibility into the developmental process, so that teams can efficiently and effectively respond to change requests.
- Engage the customer; with a focus on collaboration and emergent design, to develop a product that truly meets the customer's needs. Even if those needs should change or evolve over time.
- Collapse feedback loops for efficient time management, and effective communication.
- Co-locate the team, because this significantly collapses the feedback loop.
- Distributes technical decisions across self-organizing teams, where collective ownership, flexibility, and emergent design are valued principles that drive team behavior.

From	To
Prescriptive	Adaptive
Project	Product
Scope-boxing	Time-boxing
Individual Accountability Model	Collective Ownership
Infrequent Releases, Large Batch Sizes	Continuous Delivery
Demand Management	Dedicated Resources
Functional Silos	Cross-Functional Teams
Component-based	Feature-based

Agile tries to dispel with the heavyweight approach of traditional development methodologies. You know… those that require lengthy requirements documents, change requests for every enhancement, signatures for every decision, and contractual agreements that put the focus on what is known up front – rather than on what emerges as the team learns more. Do these things add *direct* value back to the customer? Instead, Agile places the emphasis on collaboration with the customer, welcomes change requests, and tries to focus on the most important items first, so that the product emerges in a way that is of ultimate use.

We are not building skyscrapers, or aircraft carriers. We are building software. Thus, a modular, incremental approach, allows for maximum flexibility, and leads to a culture of innovation. It is not imperative that we have all requirements defined up front, because unlike the construction industry, we don't need to build a foundation that will support the Empire State Building, or a hull that will account for 120,000 tons of water displacement. Instead, we need a technical foundation that will allow us to adapt to emergent trends in an industry that literally moves at the speed of light.

The Learning Economy

Waterfall and Agile are both tools in the toolbox, and each have their applicability. However, Information Technology — like the Healthcare industry — is part of the knowledge work economy. Knowledge work — which is characterized by heuristic tasks — exists in what is known as a complex work domain (Cynefin Framework). Our ability to accurately predict the scope for complex work, and then derive schedule estimates based on that, is severely prone to uncertainty (Cone of Uncertainty). This is further compounded

by the accelerating pace of change (Law of Accelerating Returns), where market conditions rapidly shift, and emergent technologies create disruption.

Furthermore, we develop software in accordance with the communication structures of our organization (Conway's Law). Thus, if the business is compartmentalized from our developers, the software products we create are likely to mirror that elongated feedback loop structure. This can lead to sub-par product designs.

Waterfall originated in the construction and manufacturing industries, with several contributions from the government sector (e.g., the Navy introduced PERT and the Work-Breakdown Structure in 1957). It was later applied to software development, but isn't ideally suited to the nature of this work, which is why iterative methods (such as rapid prototyping) and later Agile frameworks began to emerge, tailored and adapted specifically to the needs of the IT industry.

A report published by the Standish Group in 2004 found that only 47% of Waterfall projects meet the PMBOK definition of success based on the iron triangle criteria (completed on scope, schedule, and budget). When they included one additional criteria — arguably the most important one — and asked if the project achieved the anticipated benefits, the success rate dropped to 29%. The latest updated statistics provided by the Standish Group earlier this year shows that Waterfall has further declined as the pace of change has hastened… dropping to 11%. According to that same report, Agile projects are four times more successful.

Due to this high rate of failure, organizations often – albeit mistakenly – believe that the more time they spend in planning, the better will be the outcome. But this is seldom the case in a complex work domain, where the market conditions can change rapidly, and the consumers' needs change and evolve over time. A plan is only as good as the product it produces… and when 89% of Waterfall projects fail to deliver the intended value back to the customer, then clearly the quality of the plan does not correlate directly with the quality of the product.

While Waterfall methods often claim a tangible output at the end of each phase, the tangible output is often something that provides indirect value, rather than direct value, to the customer. For example, at the end of the requirements phase, the tangible output is a requirements document. This document often provides a false sense of assurance that we can accurately predict the needs of the customer, sometimes even years in advance depending on the length of the development cycle. A requirements document is not something that the customer can use, or that provides any value back to the business. This is known in Lean terms as indirect value. In contrast, Agile seeks to provide direct value every two weeks (or sooner) by producing working software that the business can immediately deploy and the customer can use. The goal in Agile is to focus on the most important, high value features, and deliver those first. As in marketing and business, the 80/20 rule also applies to software development (Pareto Principle), where 80% of the value is often found in 20% of the feature set. Build that 20% first, and you've delivered 80% of

the products value in 20% of the time, and deployed that valuable feature set to the end user, which substantially increases the overall ROI.

Waterfall, due to its sequential and phase-based nature, is developed in functional silos, which results in phase-gates, handoffs, and CYA behaviors. This actually conceals and hides progress status, which is why projects often go red based on lagging — rather than leading — indicators. Meanwhile, Agile frameworks work in small iterations using empirical process control, with an embedded member of the business, to create true transparency. The business is in control at all times, determining what high-value features should be worked on next. In addition, they get to see the product as it emerges, and make adjustments accordingly.

While the time investment on behalf of the business representative is significant in Agile frameworks, this level of involvement is imperative to achieving success. The business becomes the rightful owner of the products IT creates, so it's imperative to have members from the business involved throughout the development cycle. Few people can accurately communicate or envision up front what they would like the product to do or look like down the road… often people are much better at seeing something as it emerges, and then offering feedback and course correction. This keeps IT and the business aligned, and ensures our customers get something that is truly valuable.

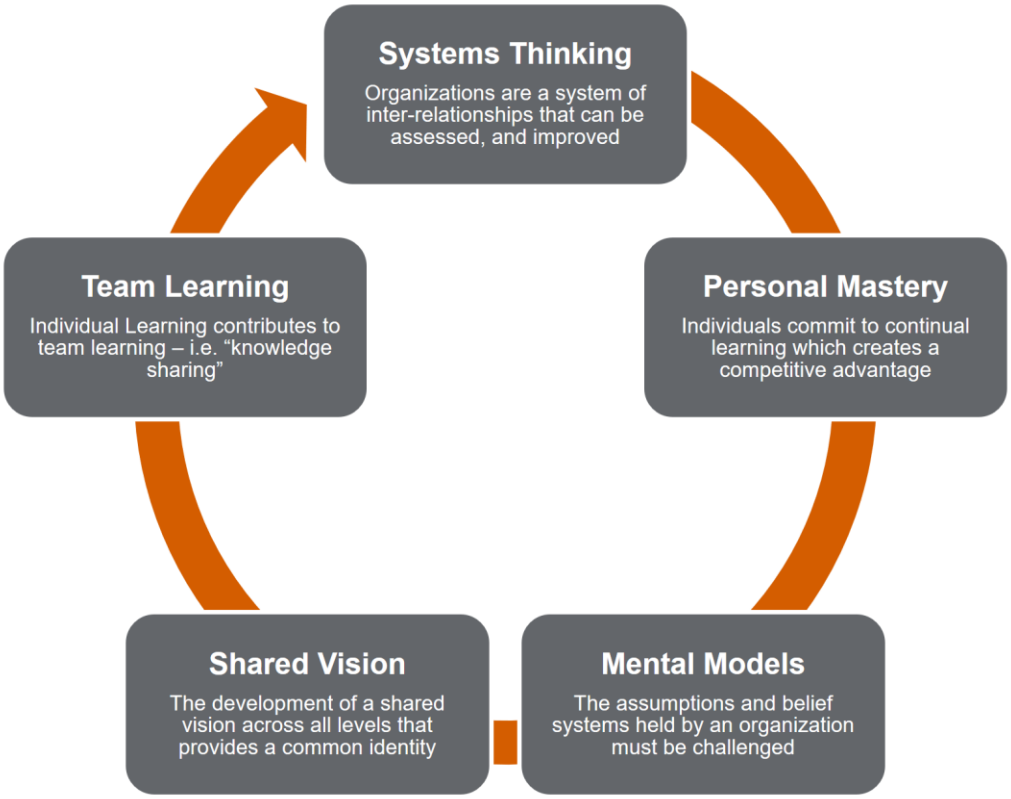
Agile frameworks are quickly becoming the standard for software development. The Project Management Institute-Agile Certified Practitioner (PMI-ACP) is the fastest growing certification in the history of the PMI, and the entire PMBOK is now being redesigned to fully incorporate Agile methods (Stenbeck, 2016). It is no longer a unique or alternative approach. A change of this magnitude — when Waterfall has been ingrained in many organizations for 70+ years — will encounter resistance. Transformation is not easy.

There is constant pressure to do more with less. In the United States, the workforce is shrinking as the baby-boom generation begins to retire. The Bureau of Labor Statistics projects that workforce growth will slow to .3% annually. Meanwhile, workforce demand in Computer and Information technology is projected to grow to 12% through 2024. The demand for IT skills is increasing, while the available workforce is shrinking. This means companies must adopt a learning organization model, where they continuously invest in and refresh the skills of their employees in order to maintain a competitive edge. Furthermore, extensive research shows that small teams are significantly more productive. Thus, while Agile can be done at scale with many teams supporting a single product… the more that work can be chunked into small batches and distributed across small teams, the more productive each member in the organization will be, and the more throughput will increase.

Agile is not a silver bullet. It doesn't tell a company what products to build, how to structure their organization, or what markets to enter. However, it does provide them with a high level of transparency, a collapsed feedback loop structure, and the ability to inspect and adapt their products as they emerge. There is constant emphasis on value delivery, and continuous improvement is built directly into the process itself. Thus, while Waterfall is a

perfectly viable method, and great things have been achieved with it, it may not be the best tool in the toolbox for software development, or any other type of complex work.

That being said, Agile no longer provides the competitive advantage it once did. The secret is out. Too many people are doing it. The goal should no longer be to merely transform to Agile, but rather to create a learning organization, defined by cultural agility, that allows your company to quickly adapt to emerging realities. The pace of technological innovation isn't going to slow down anytime soon. If you're just now thinking of transitioning to Agile, it may be too late. Some Lean startup is probably eating your lunch.



THE BENEFITS OF AGILE

Any significant change will be met with considerable resistance. It's human nature. The corporate antibodies of an organization will rise up to eliminate the foreign element. After all, the reward systems of that company were all built around the previous organizational design. The incentives are stacked against you. As for the leaders of that company, they were promoted to that position – at least in part – by proving how effective they could be in the old system. Asking them to change from something they have considerable expertise in, to something they barely understand, is not an easy challenge.

I often hear statements such as this, "Agile has never been *proven*." If I dispel this myth, then it is often followed by another misconception: "Well, Agile has never been proven in *our* industry." If we explore this further, often it turns out to be a false assumption, and then I'm confronted with one last rebuttal: "That's all well and fine… but Agile has never been proven *here*."

Every organization I've helped transform has fallen into this same trap at some point or another. They all believe they are unique, and that what works in the industry won't work for them. While I agree that every transformation has its own specific nuances, the degrees of separation aren't as stark as these companies like to believe. I see the same problems over and over again (e.g., functional silos, program/project-based funding models, legacy architectures, role definitions, misaligned incentives, space design, etc.), and in the end, it's the same solutions that fix them. But it seems like every organization has to learn these lessons for themselves. In this way, every organization is uniquely the same.

My recommendation is always for these companies to start by implementing what is generally accepted as a best practice within the Agile community. Start simple, and focus on the three components that Mike Cottmeyer suggests (i.e., form teams, create backlogs, and produce working, tested software). Create some learning milestones. Focus less on modifying Agile to fit your organization, and more on modifying your organization to fit Agile. It is, after all, a change in culture that we're implementing. If you don't do this, you'll be as likely to move in the wrong direction as down the right path. Too many people want to change things before trying it as designed. Would you cook a Soufflé for the first time while significantly deviating from the recipe, or would it be better to follow the directions, and then determine how to modify the recipe in the future? If you break the rules before you understand them, you'll end up with unintended consequences and aggravation that could likely have been avoided. When you do break the rules, do so with intent. Understand the effects these actions will have. Ask yourself the question: Am I wanting to make this change because it will make Agile more effective in my organization, or because I'm still seeing things through a Waterfall lens?

Take the advice of those who have been through this before. They have hard-earned battle scars… don't dismiss their guidance lightly. There is an old saying, "A wise man learns from his mistakes. An intelligent man learns from the mistakes of others." Most organizations are the wise man. Think of how much time your company might save if it became the intelligent man instead?

Agile *has* been proven, and odds are it's been proven in your industry. From Lean manufacturing to every top technology company out there, regulated and unregulated alike. The odds are also good that it's been proven inside of your organization, even with every upstream and downstream process fighting against it. Every transformation I've seen begins with a pilot program or two, and then begins to expand from there. It often starts as a bottom-up change, but then requires a top-down effort to break through the glass ceiling. Look around your organization… maybe there's more happening than you know.

There has been a surprising amount of research on Agile, from Carnegie Mellon University to Harvard Business Review, and many others. The results have been clear: Agile consistently outperforms Waterfall on every measure, whether you're looking at net promoter score (NPS), employee engagement, speed to market, or quality. Here's a small, comparative sampling of that research:

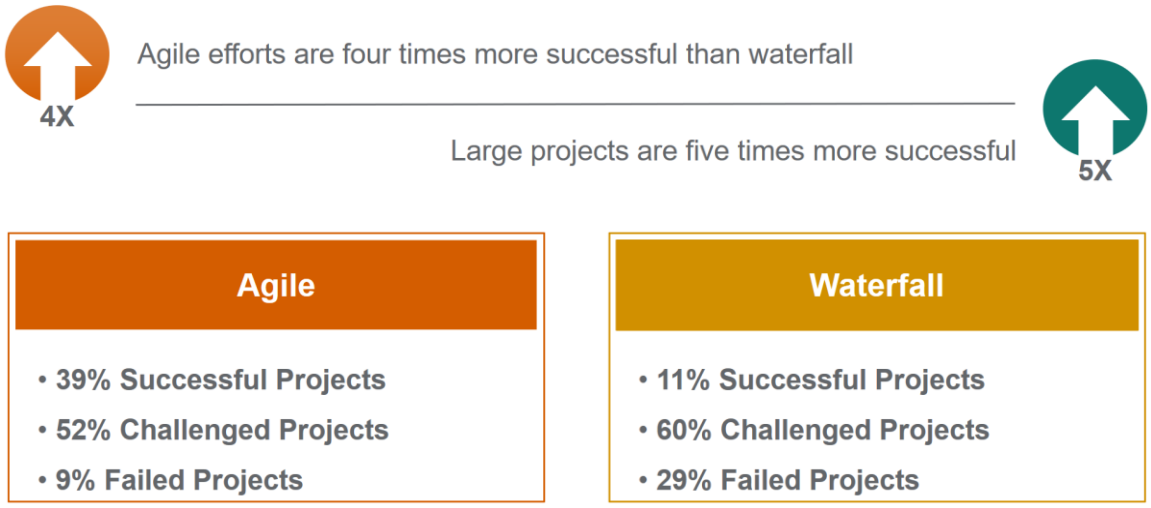
Many Companies Across Many Industries Leverage Agile

The Standish Group – 2015 CHAOS Report

The annual CHAOS report, published by the Standish Group since 1994, analyzes thousands of projects every year throughout the software development industry. The latest report examined over 50,000 projects from around the world. The research study examines many attributes – including project size, Agile versus Waterfall, and factors of success – to create a snapshot of the industry. This year's report utilized the following definition for project success:

- **On time**
- **On budget**
- **Satisfactory result**

Jennifer Lynch, a Communication Manager for the Standish Group, explained the importance of the third attribute in determining project success, "The Project Management Institute (PMI) has defined success as on time, on budget, and on target; also known as the triple constraints and the iron triangle. However, we have seen many projects that have met the triple constraints and did not return value to the organization or the users and executive sponsor were unsatisfied."



The Standish Group has identified and ranked factors that interrelate and contribute towards a projects successful outcome. These factors are listed below, along with the average impact each attribute had towards determining success.

Factors of Success	Impact
Executive Support	15%
Emotional Maturity	15%
User Involvement	15%
Optimization	15%
Skilled Staff	10%
Standard Architecture	8%
Agile Proficiency	7%
Modest Execution	6%
Project Management Expertise	5%
Clear Business Objectives	4%

The description for each factor is listed below:

- **Executive support**
  - The project has financial and emotional backing from executive leaders.
- **Emotional maturity**
  - The collection of basic behaviors that govern how people work together.
- **User involvement**
  - Users are involved throughout the project to make decisions, review the product, and provide feedback.
- **Optimization**
  - A structured means of improving business effectiveness through management of scope based on business value.
- **Skilled staff**
  - The people on the project have the skills necessary to execute against the deliverables, and understand both the business and the technology.
- **Standard architecture**
  - There exist a defined and integrated group of practices, services, and products that are consistently applied for developing, implementing, and operating software applications.
- **Agile proficiency**
  - All members of the Agile team are skilled in executing the Agile process.
- **Modest execution**
  - The process should have few moving parts, and those that exist should be streamlined and automated, with project management tools used only sparingly.
- **Project management execution**
  - Knowledge, skills, and techniques that are applied to project activities in order to meet stakeholder expectations.
- **Clear business objectives**
  - The project aligns to the organization's goals and strategic vision, the business purpose is clearly defined, and all stakeholders and participants are understood.

The above factors and corresponding impact to project success make a very compelling case for Agile. For example, Agile utilizes a flatter organizational structure to ensure *executive leaders* are as close to the work as possible, advocating for information radiators that allow management to wander through the team spaces and immediately see the status of all project efforts. Agile creates *emotional maturity* by defining working agreements for each team, and provides the teams an opportunity to increase their maturity over time through the use of retrospective ceremonies, static team membership, and collapsed feedback loop cycles. No other method advocates for *user involvement* to the extent that Agile does, where the Product Owner is often embedded directly in the team, and all stakeholders are invited to review the product as it emerges. Meanwhile, *optimization* as described above is almost entirely an Agile construct, where the requirements that have been defined on the product backlog are continuously reprioritized by the Product Owner to ensure value delivery. *Skilled staff* is another way of saying that the members of the team are cross-functional, which means they have all the skills necessary to deliver a working

product increment. *Standard architecture* means that there is a defined and consistently applied organizational method to how we approach product development. *Agile proficiency* means that if we are using Agile, then those on the team understand what that is and how to apply it. Furthermore, it's important we have *modest execution* to keep our processes lightweight, fast, responsive, and to limit the potential for error. *Project management execution* is the knowledge and skill to apply the right technique at the right time. *Clear business objectives* – while important – sits at the bottom of the list… as it is more important that our objectives align to the organization's goals and strategic vision, but less important (and actually prohibitive) if we use a big design up front (BUFD).

The report concluded that, "Across all project sizes, Agile approaches resulted in more successful projects and less outright failures."

Case Study Results for Scaled Agile Framework

The Scaled Agile Framework has compiled data from numerus case studies, and the results closely match the large body of independent research that has been published on Agile. Empowered knowledge workers tend to be happier, more motivated employees. An increase in motivation typically translates into an increase in productivity. Happier and more productive employees produce a higher quality product. Plus, *how* we develop a product is as important as *what* we develop. A more productive workforce, that increases quality by keeping their technical debt low, will increase their speed to market.

Department of Defense – The Business Case for Agile Methods

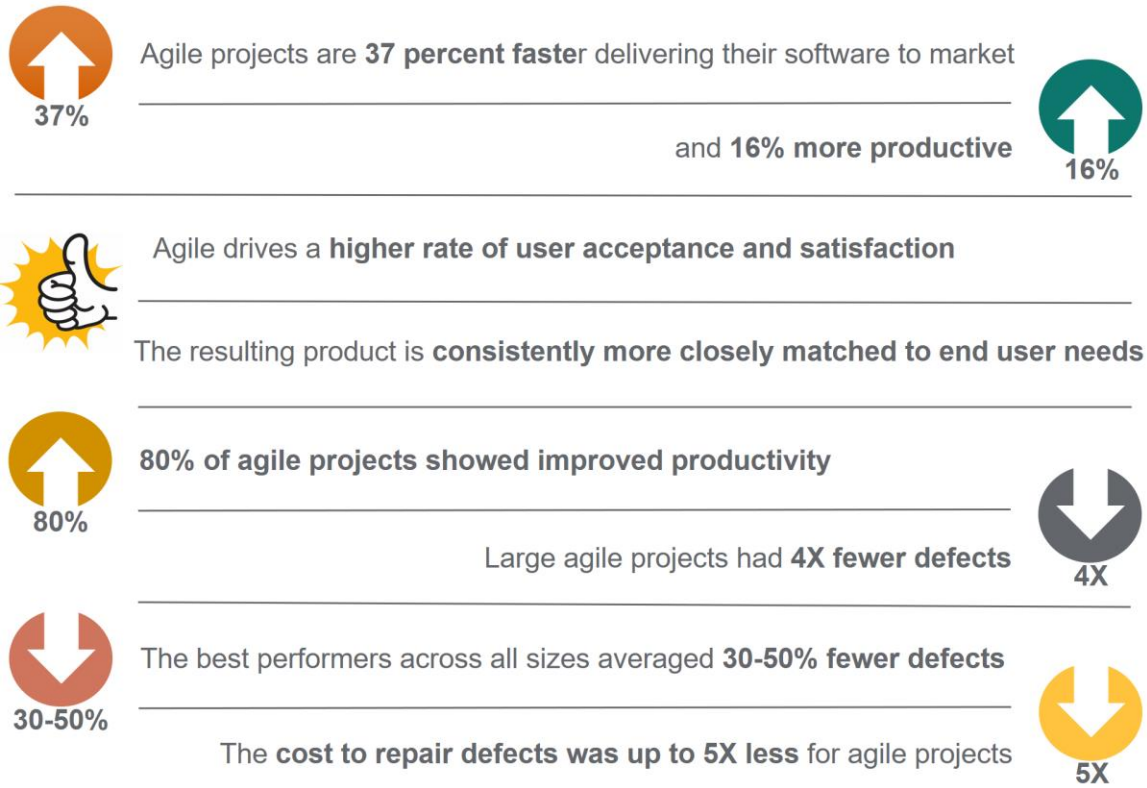
In 2011, the Department of Defense (DoD) commissioned a report from the Association for Enterprise Information (AFEI) on the business case for Agile methods. The authors included members of Northrop Grumman, Excella Consulting, Rally Software, Asynchrony, and Jacobs/Tybrin; along with contributions from members of the Naval Post Graduate School, Boeing, Software Engineering Institute, and the Stevens Institute. Not

only did the authors look at the extensive amount of research that has been documented on Agile within the commercial industry, but they also provided numerous case studies from within the Department of Defense itself that showed similar results.

The conclusion of the report read as follows:

> *Improved productivity, quality, time-to-market, and customer satisfaction are common goals of the DoD and commercial sector. Industry has demonstrated that the implementation of Agile practices can produce significant results in these areas.*

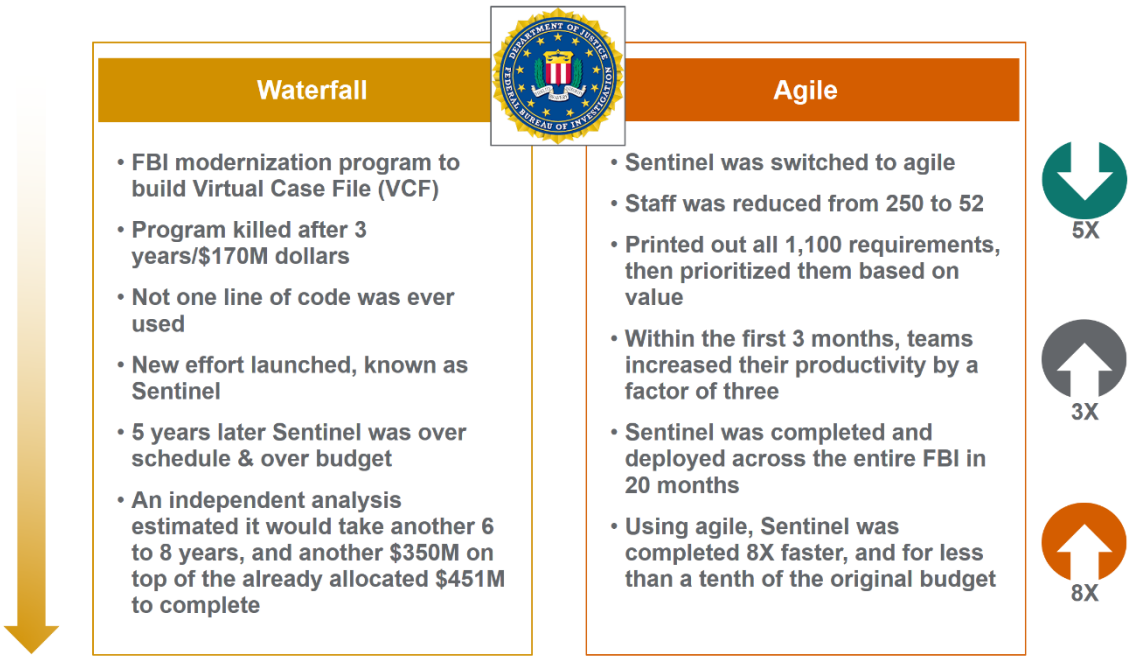
The following ROI benefits were listed in the body of the report:



Federal Bureau of Investigation – Sentinel

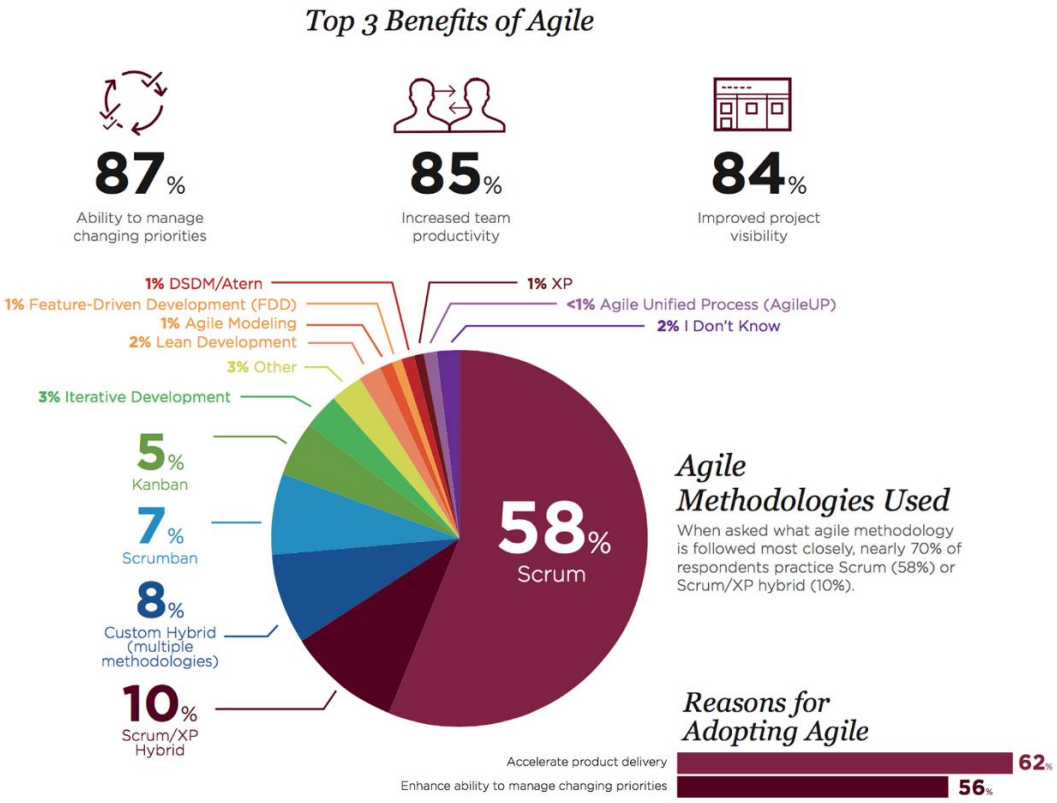
In Scrum: The Art of Doing Twice the Work in Half the Time, Jeff Sutherland relays the story of Sentinel, a project launched by the Federal Bureau of Investigation (FBI) that initially used a Waterfall method, only to switch to Agile when cost and schedule overruns began to threaten the jobs of all those involved. Sentinel provides an excellent case study, because it shows us a side-by-side comparison of a project that was launched (twice no less) using Waterfall, then scrapped and relaunched using Agile methods. Thus, we get a true apples-to-apples comparison, and the results were astounding. Using Agile, not only did Sentinel get deployed across the entire FBI system within 20 months of relaunching, but it was completed eight times faster, with five times fewer staff, at a tenth of the original budget.

Delivering Value: The Benefits of Cultural Agility



Version One – 2016 State of Agile Report

Version One has released the results of their large-scale Agile survey every year for the last decade. The 2016 report contained a wealth of information. Here are a few highlights:

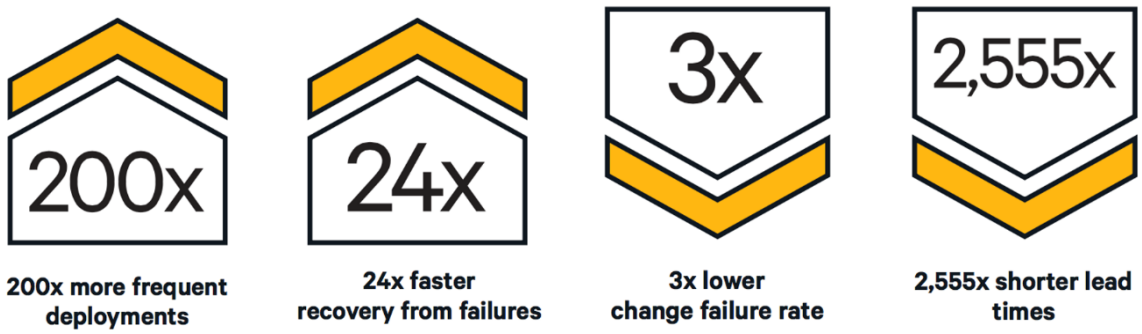


Scrum Alliance – The 2015 State of Scrum Report

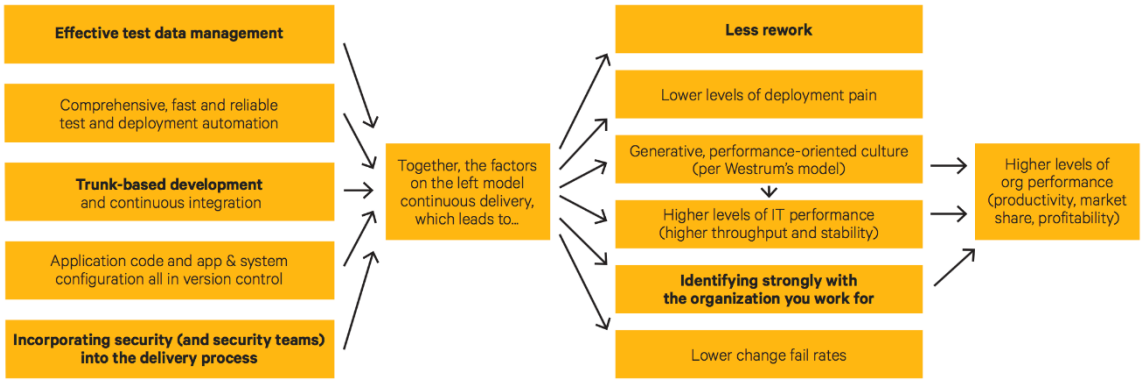
Similarly, the Scrum Alliance produces a biennial report. A few notable statistics are listed below, which demonstrate the growth and global coverage of Scrum.

Puppet – 2016 State of DevOps Report

Over the course of the last five years, Puppet – in conjunction with a myriad of partners – has released an annual report on the state of DevOps. While there is some disagreement on the true definition of DevOps, it can generally be considered a set of technical practices and collaboration techniques that fit under the umbrella of Agile. With nearly 5,000 respondents participating in this year's survey alone, the benefits for those who have implemented DevOps have been clear:

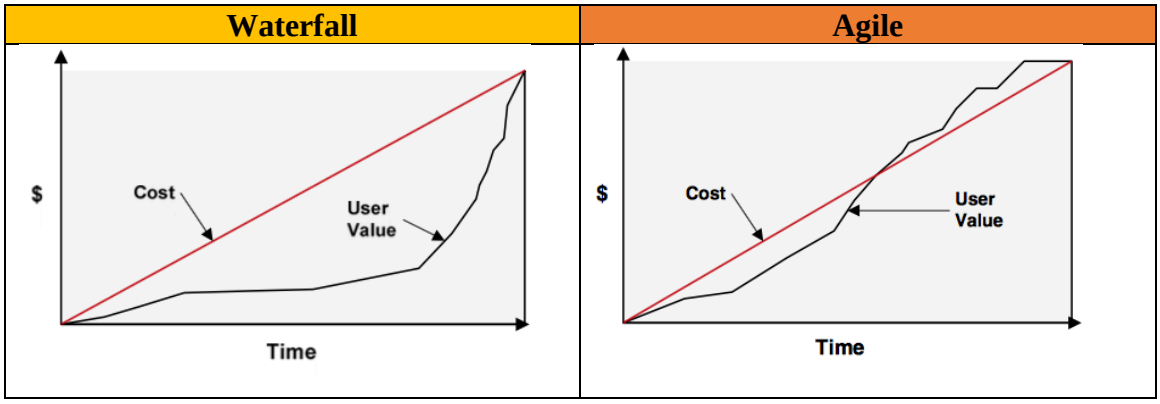


The authors of the report included a diagram of the inputs and outcomes in order to further demonstrate the benefit realization of DevOps. Organizations that implemented continuous delivery (CD) had less rework, easier deployments, a performance-oriented culture, higher levels of IT performance, increased employee engagement, and lower change fail rates. In turn, this resulted in increased productivity, market share, and profitability.



Carnegie Mellon University – Value Curve & Test Effort Comparison

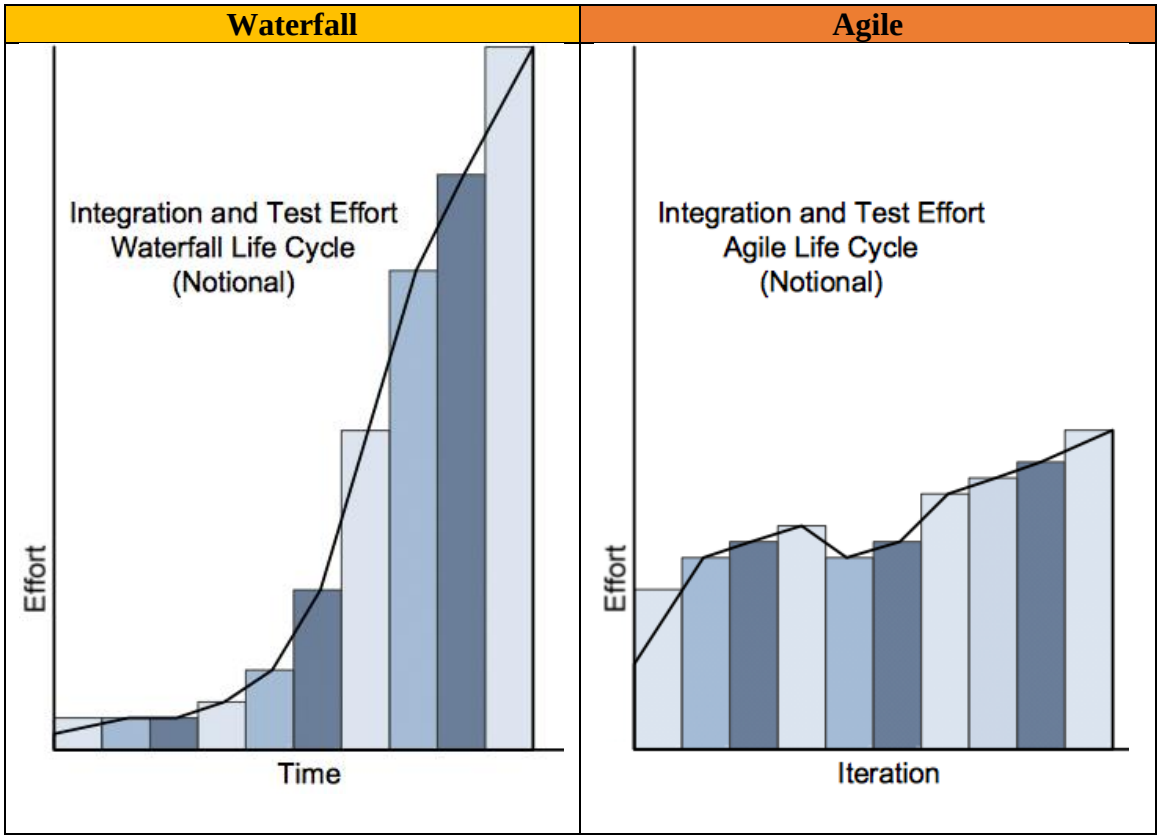
The Software Engineering Institute at Carnegie Mellon University is a federally funded program that has done extensive research on Agile and Waterfall methods. In multiple comprehensive studies, SEI demonstrated the different value curves associated with each method. In the case of Waterfall (depicted below on the left), very little value is delivered throughout the development effort until the very end, when the project is deployed to production and available to the end users. In contrast, the value curve on an Agile project (right) begins much earlier in the development lifecycle, as these teams rely on iterative and incremental delivery to provide a continuous stream of value to the end user, and solicit feedback along the way. Due to the feedback loop and the ability to adjust course as the product emerges, Agile projects often exceed the original value estimates, and they do so with much less inherent risk throughout the effort.



Furthermore, since testing is a required component of every iteration in Agile, the impact this has on the end product, from both a design and quality perspective, is quite significant. SEI states that, "Because Agile forces an executable product to be produced, the Agile team learns about integration and testing issues very early in the project. Lessons learned from early integration and testing will influence future iterations."

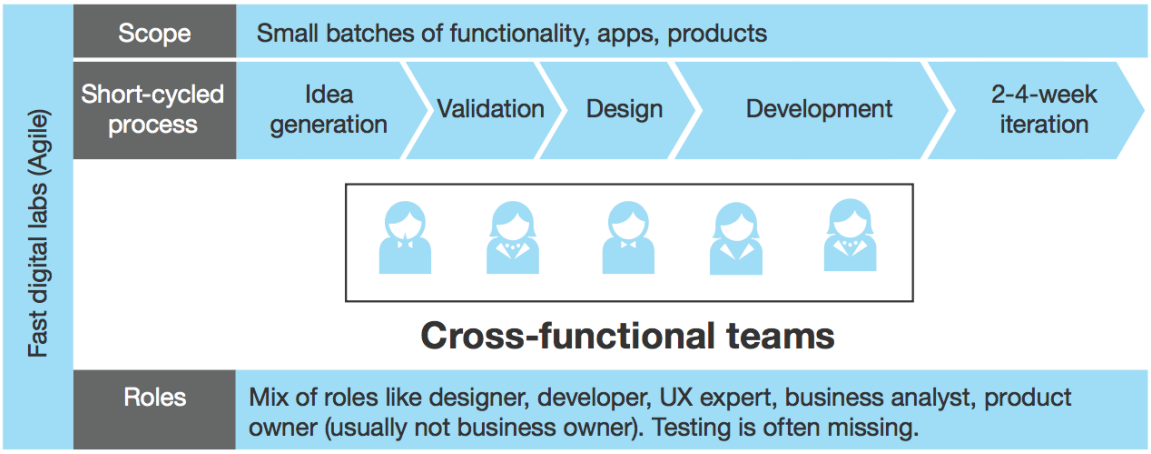
Below is an image of the test effort comparison between Waterfall and Agile. Because Waterfall uses a sequential, phase-based approach that commonly saves all testing until the end of the development cycle, the effort to test the product immediately before it deploys

is significant, and errors found at this late stage are often costly to correct. In contrast, Agile tests the code throughout the entire development effort, and thus has a more predictable pattern that results in higher levels of quality. Because testing is done soon after the code is developed, very little has been built on top of it, and changes are relatively easy (and low cost) to make. The SEI therefore concluded that, "The Agile workforce seems to have a more stable distribution of effort among the various disciplines than when using Waterfall."

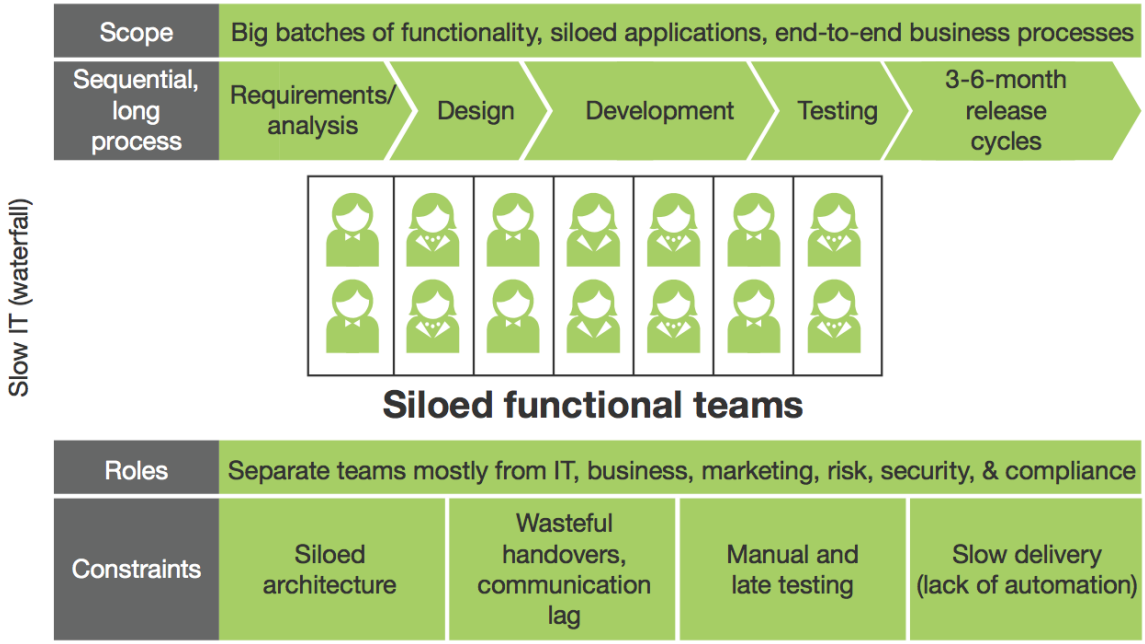


Forrester – The Need for an End-to-End Agile and DevOps Environment

The research and advisory firm Forrester recently published a study on companies that used a combination of Agile and Waterfall delivery within the banking industry. While the configuration of their Agile teams was not ideal, they were still mostly cross-functional. This article is particularly fascinating, because it looked at the effect one part of the organization had on the other. In this case, the legacy systems used a Waterfall method, while the modern technology platforms used Agile. The total level of agility for the organization as a whole was hampered, because the system did not improve as a whole. Thus, to gain the full benefits of Agile, you must look at doing an end-to-end implementation.

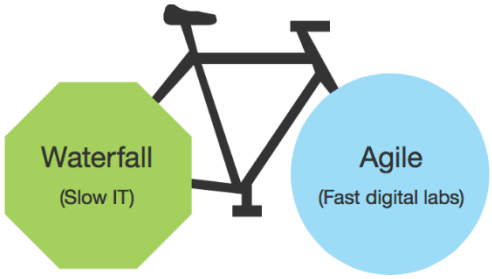


In contrast, the Waterfall teams had to deal with considerable constraints due to their siloed functional structure:



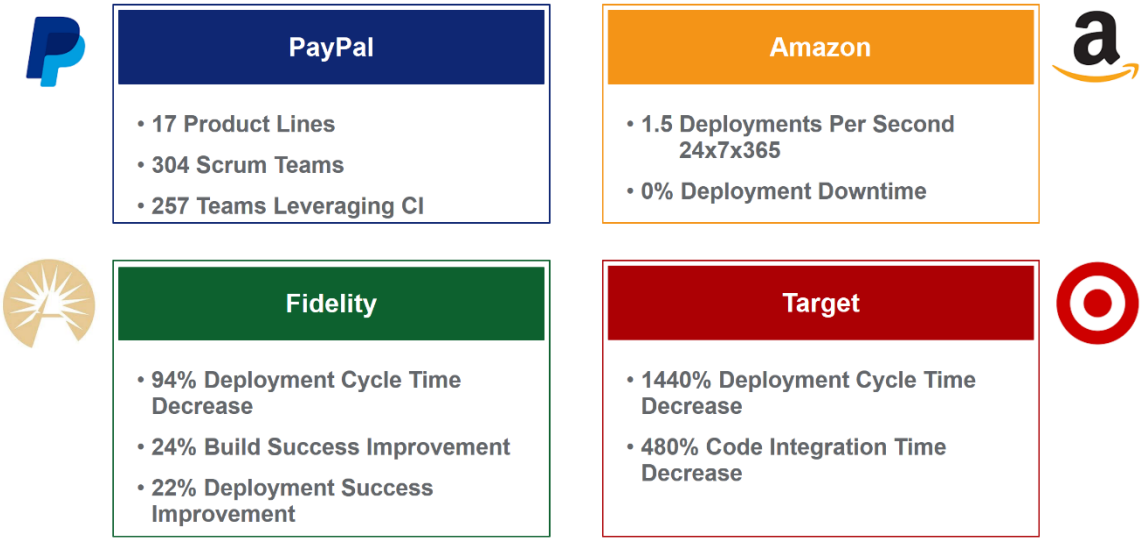
Recently, many companies have attempted to implement hybrid models. One such model that has been gaining a lot of traction in the industry was developed by Gartner, and is known as Bimodal. Many consider this to be an evolution of a previous model called two-speed architecture. Essentially, the company separates their legacy systems from their modern technology platforms, and then proceeds to use Agile on the modern side of the house, while continuing to use Waterfall on the legacy side. This is a dangerous scenario, because it simply kicks the can of technical debt down the road, and prevents an Agile culture from ever becoming established. As the Forrester report concluded, “Two-speed architecture is an admission of failure to transform.”

Eliyahu M. Goldratt taught in his book The Goal that, "There is a flaw in measuring efficiency of a part of the process. Just because one step does things quickly might not matter. What matters more is throughput." Forrester depicted this concept as a bicycle. One wheel is round (i.e., the Agile part of the organization), while the other wheel is an octagon (i.e., the legacy Waterfall side). In the end, both wheels are still attached to the same bike frame. The implication here is that we can only go as fast as our slowest dependency.



Agile and DevOps Success – By the Numbers

The Agile and DevOps movements have created thriving communities that love to share what they've learned. Where companies once may have considered their technology implementations to be a trade secret, they now share their success stories openly via user meetups, webinars, and conferences. Here are some of the benefits that other companies have self-reported, based on their own Agile transformations:



While Agile often outperforms Waterfall, even when done haphazardly, many companies fail to fully exploit the benefits that Agile can bring to their organization. This happens for numerous reasons, but the biggest cause is often a failure to commit fully. Agile is seen as a process, not a mindset, and thus it never gets embedded within the company culture. It becomes a way of *doing*, not a way of *being*… and this is the surest sign that you're probably not getting everything you could out of your transformation.

To get the most out of Agile, your organization should focus on implementing a team-centric design. Previously, most Waterfall organizations were manager-centric, in which his or her needs were prioritized above all else. The organizational design was structured in such a way as to maximize their individual efficiency, rather than the efficiency of the team as a whole. She or he would emerge from their office, declare a new top priority, and then look for the status report on their desk the next morning. The teams would prepare elaborate presentations and project reports to make sure the manager was kept informed. Sometimes, the team would spend more time preparing the report or the presentation than they did working on the actual product or service. Lots of emphasis was placed on developing indirect, rather than direct value, because it seemed more efficient for management. This shifts in an Agile environment, and it's the result of Lean thinking.

No longer is the manager the epicenter. Now, the team is. This is because the team does the hands-on work of developing a product or service that gets delivered to the customer. While the manager defines the strategic vision, assembles and guides the team, and 'commands the ship', the team members are the ones who actually *row*. If you want them to go faster, then make sure they are cared for and have everything they need. Remove obstacles from their path. Would a better oar help? Perhaps we should upgrade the technology platform… what about a nuclear powered engine? Provide the team with a well-crafted strategic vision, as the purpose-motive is as important as the profit-motive, if not more so.

In a team-centric design, the manager does everything they can to make the team more efficient. How might a manager help their teams be more productive? One technique in Agile is to limit the number of interruptions the team has. In Scrum, the team does not attend any other meetings beyond the designated Scrum events (i.e., sprint planning, daily Scrum, sprint review, and sprint retrospective). The reason for this is simple: the team is co-located and cross-functional, so everything they need to deliver a working and tested product increment should already be at hand. Working on the product will provide direct customer value. Attending a meeting will provide indirect value at best, and often none at all. To develop and deliver working software in two weeks or less, it requires dedicated focus, and heads-down time to work. Don't pull each person away individually to get a separate update on the progress of the work. Instead, visit the team space. Look at the team's information radiator, or check the system dashboard to see real-time data. Attend the sprint review. Conform to the team's patterns of efficiency… not your own. Become a servant leader. You'll be surprised how much efficiency increases.

SERVANT LEADERSHIP

All Agile frameworks use a servant leadership approach, which is different from a traditional, command-and-control orientation. (Greenleaf, 1970). The adaptive planning techniques of Agile are ideal for a complex work domain, such as software development, which exists as part of the knowledge work economy. Knowledge workers are highly skilled specialists who know a great deal about the work and the environments in which they perform. We can increase speed and efficiency by decentralizing certain types of decisions and empowering those throughout the lowest levels of the organization.

Methods of Leadership

A servant leader embraces the democratic method of leadership. Autocratic leaders are exclusive. They use top-down, command-and-control methods. Their actions are based on the notion that management knows best, and therefore does not need to consult with subordinates to inform or refine their decisions. Democratic leaders are inclusive and paternalistic. They welcome input from subordinates. They base decisions on what is best for both the organization and its employees. They nurture subordinate growth and professional development.

From	To
Command-and-Control	Servant Leadership
Autocratic, Exclusive	Democratic, Inclusive, Participatory
Micro-management	Empowered Teams
Manager Centric Design	Team Centric Design

McGregor's Theory of Motivation

McGregor categorizes managers according to the two belief systems (Theory X and Theory Y) that characterize their approach. Individuals who subscribe to Theory X believe subordinates are inherently opposed to work and will seek to avoid it if possible. They believe employees fear taking responsibility, crave security, and badly need direction. As a result, they think "most people must be coerced, controlled, directed, and threatened."

Individuals who subscribe to Theory Y believe subordinates are motivated and self-directed and their talents are underused in most organizations. They infer that under the proper conditions, people will accept, and even seek, responsibility.

McGregor felt the traditional model (Theory X) was based on an incorrect set of assumptions, and that Theory Y more accurately assessed the human condition. He felt that Theory Y was a more effective starting point for running companies. In The Human Side of Enterprise (1960), McGregor argued that if your company's starting point was Theory X, then your managerial techniques would inevitably produce limited results. But if your starting point was Theory Y, the possibilities were vast… not simply for an individual's potential, but also for the company's bottom line.

A servant leader embraces Theory Y.

Pink's Reinterpretation

In Drive (2009), Daniel Pink argues that many organizations *claim* to have shifted from Theory X to Theory Y, but this is simply not true. What has become well understood in the classroom and is now supported by extensive research, has flown under the radar in the boardroom. Pink says that there's a mismatch between what science knows and what business does. A servant leader understands and enables Type I behavior.

Type X behavior is characterized by command-and-control, carrot-and-stick motivators, as well as motivation principles formulated during the industrial age. These fail to deliver results in the knowledge age, and often actually decrease productivity, rather than increasing or maintaining it. The extrinsic rewards found in Type X can deliver fast results, but this approach is not sustainable and soon leads to employee burnout.

According to Pink, Type I behavior is characterized by the presence of:

- **Autonomy**
  - Our desire to be self-directed.
- **Mastery**
  - Our urge to get better and better at what we do.
- **Purpose**
  - Our yearning to be a part of something larger than ourselves.

Type I motivated individuals are fueled more by intrinsic desires rather than extrinsic ones. They are less concerned with the external rewards to which an activity leads, and more concerned with the inherent satisfaction of the activity itself. This type of orientation is independent of age, gender, or nationality because it arises in part from universal human needs. Type I individuals aren't directly pursuing conventional notions of success. They work hard and overcome obstacles based on their desire to learn, to grow, to control their own lives, and to do something meaningful.

Task Complexity

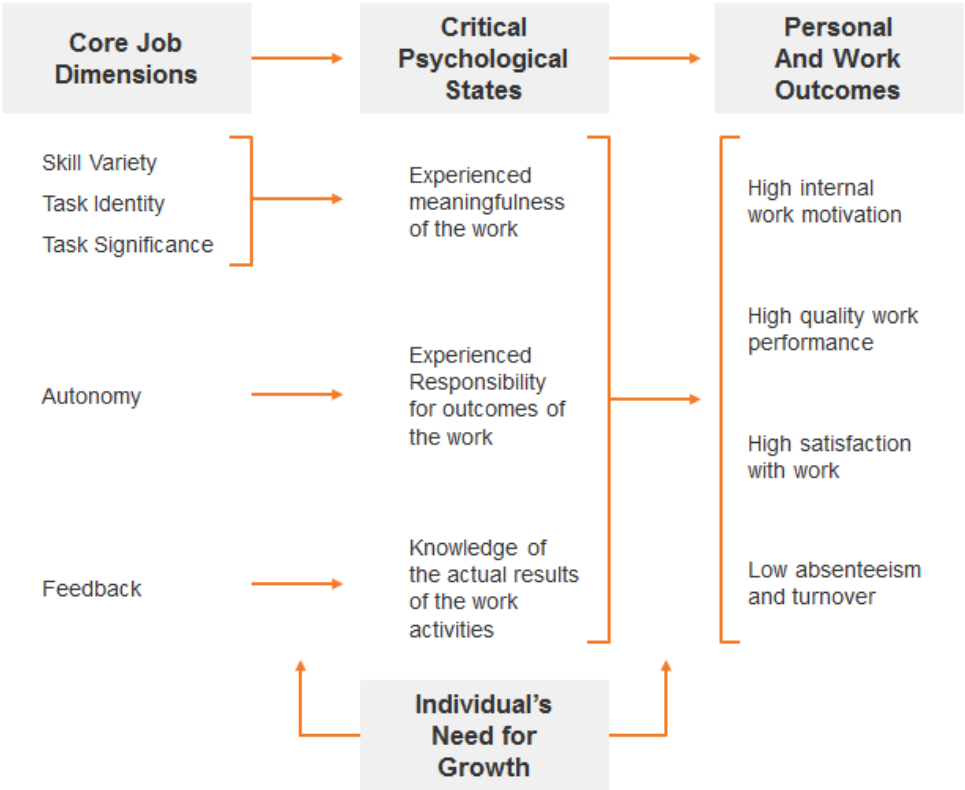
Most knowledge work is characterized by heuristic, rather than algorithmic, tasks. Heuristic tasks require that workers experiment with possibilities to find a solution. Algorithmic tasks follow a set of established instructions down a pathway to one conclusion. According to Contingency Theory, algorithmic task work is found in *mechanistic* organizations, but *organic* organizations rely on heuristic task work to gain a competitive advantage. Only algorithmic tasks are good candidates – if such a thing can even be said – for a command-and-control approach. Meanwhile, all heuristic work is better suited to a servant leader orientation.

Job Characteristics Theory

By using five job characteristics, managers can enhance three critical psychological states. These states, in turn, can improve a variety of personal and work outcomes. The psychological states that result from an increase in these five characteristics are:

experienced meaningfulness of the work, *experienced responsibility for outcomes of the work*, and *knowledge of the actual results of work activities*. A servant leader seeks to incorporate and accommodate these five characteristics for their employees. The five job characteristics are:

- **Skill Variety**
  - The degree to which the job requires a variety of activities that involve different skills and talents.
- **Task Identity**
  - The degree to which the job requires completion of a "whole" and an identifiable piece of work; that is, the extent to which a job has a beginning and an end with a tangible outcome.
- **Task Significance**
  - The degree to which the job affects the lives or work of other people, both in the immediate organization and in the external environment.
- **Autonomy**
  - The degree to which the job allows the individual substantial freedom, independence, and discretion to schedule the work and determine the procedures for carrying it out.
- **Feedback**
  - The degree to which the job activities give the individual direct and clear information about the effectiveness of their performance.



Organizational Behavior

Poor managers often rule through fear, while a servant leader will inspire others to become the greatest version of themselves. Much of this depends on the type of culture the leader's actions enable. Aspects of culture include:

- **Psychological Contract**
  - A person's overall set of expectations regarding what he or she will contribute to the organization and what the organization will provide in return. One specific aspect of managing psychological contracts is management of the *person-job fit*. A good person-job fit is one in which the employee's contributions match the inducements the organization offers.
- **Organizational Citizenship**
  - This is the extent to which the individual's behavior makes a positive overall contribution to the organization. The social context, or work group, in which the individual works must facilitate and promote such behaviors. The organization itself, especially its culture, must promote, recognize, and reward these types of behaviors if they are to be maintained.
- **Empowerment**
  - The process of enabling workers to set their own work goals, make decisions, and solve problems within their sphere of responsibility and authority. This includes the work itself, work context, and work environment.
- **Personal Safety**
  - The act of creating a safe environment for employees to contest and debate the ideas of others, including those in leadership positions, in order to improve upon the original idea. Participants must feel safe to express their opinions, and know that their involvement with the group will not be jeopardized by doing so. This leads to a *culture of innovation*, because it allows people to challenge preconceived notions and thereby elevates the level of discussion through these crucial conversations.

Management by Wandering Around (MBWA)

A servant leader keeps their finger on the pulse of their team at all times.

Hewlett-Packard pioneered "Management by Wandering Around (MBWA)" in order to stay in touch with team efforts and employee concerns. This means managers do not sit in

their offices and read status reports all day, but get out and interact with the people who are doing the work. This approach has widely been adopted in the Agile community, as teams will use information radiators or task boards to display the status of their work within the team space. Interested stakeholders can walk by at any time, immediately understand the status of the work, and speak with the team to stay in tune with their efforts. This increases transparency and decreases the us-vs.-them dynamic. This also decreases bureaucracy and barriers to communication that prevent those at the bottom of an organization from speaking to those at the top. It creates the sense of a flat hierarchy, where everyone in the organization is working together to achieve some goal greater than themselves. When senior leadership interacts with subordinates on a personal level and on a frequent and continual basis, this increases the feeling of organizational citizenship. Employees begin to feel truly cared for by their management staff, satisfaction goes up, attrition goes down, and the level of transparency continues to grow as trust builds.

Knowledge Workers

A servant leader leverages the collective expertise of everyone on their team to gain a competitive advantage.

Griffin and Moorhead write in Organizational Behavior: Managing People and Organizations (2012) that:

> *Traditionally, employees added value to organizations because of what they did or because of their experience. However, during today's "information age," many employees add value simply because of what they know. How well these employees are managed is seen as a major factor in determining which firms will be successful in the future. They often believe they have the right to work in an autonomous fashion, and they identify more strongly with their profession than with any organization–even to the extent of defining performance primarily in terms recognized by other members of their profession.*
>
> *As the importance of information-driven jobs grows, the need for knowledge workers will grow as well. But these employees require extensive and highly specialized training, and not everyone is willing to make the human capital investments necessary to move into these jobs. In fact, even after knowledge workers are on the job, retraining and training updates are critical so that their skills do not become obsolete. It has been suggested, for example, that the "half-life" for a technical education in engineering is about three years. Further, the failure to update the required skills will not only result in the organization losing competitive advantage but will also increase the likelihood that the knowledge worker will go to another firm that is more committed to updating those skills.*
>
> *Compensation and related policies for knowledge workers must also be specially tailored. For example, in many high-tech organizations, engineers and scientists have the option of entering a technical career path that*

> *parallels a management career path. In addition, in most firms employing these workers there has been a tendency to reduce the number of levels of the organization to allow the knowledge workers to react more quickly to the external environment by reducing the need for bureaucratic approvals.*

Leading by Example

Servant leaders aren't afraid to roll up their sleeves and work alongside the team, or to be seen as "part of the team" rather than "above the team".

In That's Not in my American History Book: A Compilation of Little-Known Events and Forgotten Heroes, Thomas Ayres relays the following story about General Israel Putnam, who is most famous for the line "Don't fire until you see the whites of their eyes," but who also exemplified the servant leader approach:

> *Putnam was inspecting the work of troops digging fortifications at Bunker Hill when he spotted a large stone and asked a soldier to place it on the wall. The soldier's back stiffened and he protested. "Sir, I am a corporal!" he pointed out. "I ask your pardon, sir," the general said as he dismounted. Putnam picked up the rock and placed it on the wall himself, to the delight of the men in the ranks and the embarrassment of the corporal.*
>
> *Following the battle at Bunker Hill, Washington gave Putnam the Long Island command. Among the troops eventually placed under him were smartly dressed and highly disciplined Hessian soldiers. By contrast, Putnam wore a uniform of homespun cloth because that is what the Colonials under him wore. He also ate what they ate and slept where they slept. To the Hessian troops from Europe under his command, Putnam did not look or act like a general and the old man became the butt of their jokes. One soldier would write of him, "This old gray beard might be a good, honest man but nobody but the rebels would make him a general."*
>
> *Attitudes changed when the shooting started. Putnam had seen battles the young Hessians could not even imagine, and his coolness in combat quickly won them over. After seeing Putnam under fire, the same Hessian soldier would grudgingly write of him: "He seems totally unfit for anything except fighting."*
>
> *It was a tribute that, no doubt, would have made Israel Putnam proud.*

A servant leader will take their own medicine. They can often be found sitting in the team space, rather than isolated in their office. They engage with the team and partner with them. They have the same working conditions and the same environment as the team. Policies that apply to the team apply to the servant leader, with no exceptions.

In the book Be Here Now, Baba Ram Das (Dr. Richard Aplert) writes:

> *A woman once came to Mahatma Gandhi with her little boy.*

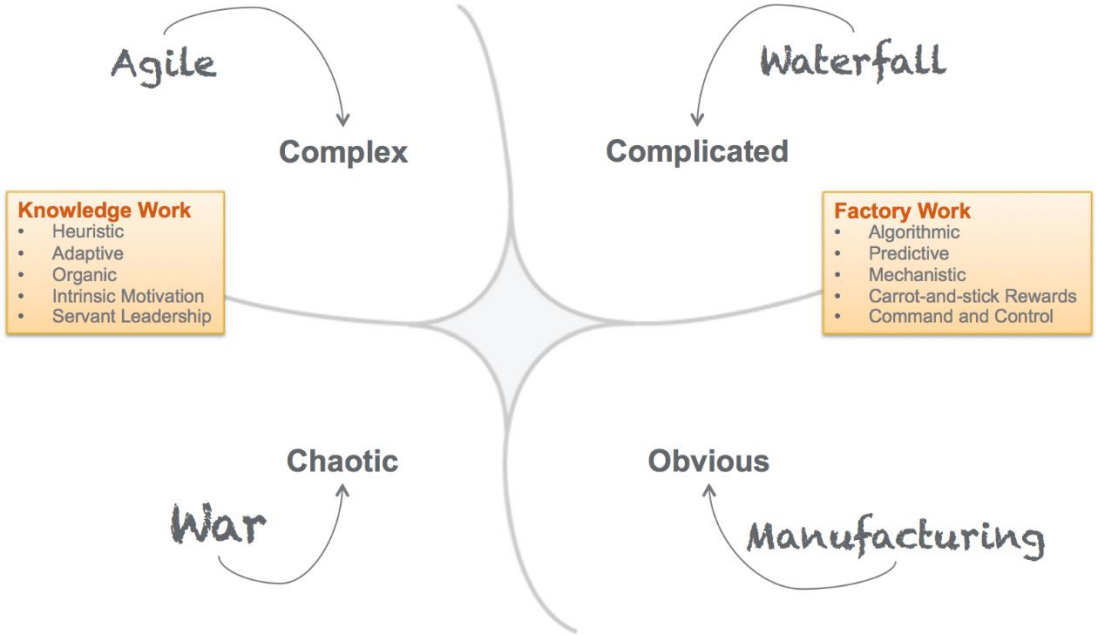
> *She asked, "Mahatma-ji, tell my little boy to stop eating sugar."*
> *"Come back in three days," said Gandhi.*
> *In three days the woman and the little boy returned and Mahatma Gandhi said to the little boy, "Stop eating sugar."*
> *The woman asked, "Why was it necessary for us to return only after three days for you to tell my little boy that?"*
> *The Mahatma replied: "Three days ago I had not stopped eating sugar."*

To be believed, the servant-leader approach requires authenticity.

Serving the Team

Agile frameworks promote an organic structure, in which tasks are adjusted as required, there is less adherence to formal authority and rules, and there is decentralized knowledge and control. Organic structures promote bottom-up efficiencies and innovation, rather than top-down directives and control. Knowledge workers respond better to intrinsic motivators such as those found in organic structures. This has been the predominant shift in the workforce as we've moved from the industrial age to the information age, from factory work to knowledge work, and from an extension of the machine to collaboration and innovation.

Traditional managers may implement policies that roll downhill. A servant leader, in contrast, manages up and out, not down. They buffer and protect their team. They teach others in the organization how to effectively interact and work with their team without crippling the team's autonomy or motivation. They serve the team, rather than themselves. Most of our knowledge work economy sits firmly in the complex work domain of the Cynefin Framework, but we still manage our employees and the work they do as though they were in a factory. The world has changed. It's time our management styles change along with it.



THE EXISTENTIAL PERSPECTIVE

People do their best work when they are supported, valued, and seen. The word *namaste* – which is used as a respectful Hindu greeting – loosely translates into English as, "I recognize the self in you." We should endeavor to recognize the self in others.

Companies must begin to see employees as human beings, rather than human resources. People aren't cogs in a machine, and shouldn't be treated as such. Each person has their own individual dreams, hopes, and aspirations. The more the company can align the goals of the individual with the goals of the organization, the more productive each will be. To see people as individuals, it helps to take a phenomenological perspective, and develop what is commonly referred to as emotional intelligence. An understanding of existentialism will help to empathize with people and the things that they value. By acknowledging the human condition, we share in our collective humanity.

The Human Condition

Existentialism is the development of self-awareness through close examination of the human condition. The human condition refers to events that are biologically determined and common to most human lives. These can include, but are not limited to: birth, childhood, adolescence, work, love, reproduction, aging, and death.

Humans are – at least to some degree – self-aware. We are aware of time's arrow (the fact that time moves in only one direction). Often, we feel the future rushing into our present. Trapped by time's arrow, we feel the weight of the human condition. This anxiety gives rise to the big philosophical questions: What is the meaning of life? Why am I here? Who am I? These are important questions, the answers for which influence the choices we make. This is how we determine who to marry, where we should settle down, what occupation we want, which company we should work for, the hobbies we develop, and what social circles we partake in. In short, these questions are at the root of identify formation.

Irvin Yalom separated the plight of the human condition into the following four categories: death, freedom, isolation, and meaninglessness. These classifications were largely influenced by Paul Tillich's 1952 seminal work, The Courage to Be, which meant the courage to live an authentic life. For Tillich, having the courage to live an authentic life meant having the courage to face the reality of non-being. Doing so meant the confrontation of the three anxieties: fate and death, emptiness and meaninglessness, and guilt and condemnation. The anxiety that results from the awareness that our time is limited creates a sense of urgency to become the best, truest version of ourselves.

Being

The concept of 'being' has undergone extensive analysis throughout the history of philosophy, beginning with Aristotle, who defined 'being' as an attribute that does not change the physical properties of an object, but rather defines that object in reality. Aristotle believed that all things which exist in reality, meaning all things that were not transcendent, had within them an 'essence' of themselves. The 'essence' of a thing was

not changed by its form (which is to say that a tree and a wooden bed maintain the essence of the wood, even if the bed does not maintain the form of the tree). In stating this, Aristotle formulated his idea of 'soma' (translated from Latin as 'soul'), as the "essence of a thing", or the "nature of something". In this way, he explained that all things human, held within them the essence of humanity, which according to Aristotle, was the ability to reason. Aristotle also said that anything which can be done is done according to natural law. Therefore, nothing which can be done can be 'unnatural', or "against nature," for the essence of a thing remains unchanged.

In A Dictionary of Philosophy by Antony Flew, we find the following definition for *being*:

> *Usually equivalent in the verbal sense to 'existence' (see is). As the most general property of all reality this is often considered to be the defining subject of metaphysical enquiry.*

Agile is also fond of utilizing the concept of being as the ultimate goal for an agile transformation. Agile coaches have long borrowed the Japanese martial arts concept of Shu-Ha-Ri to express this, which describes the transformation journey from apprentice to master. The student begins at *Shu*, where they mirror their teacher to *learn* the forms. Like a child who studies and emulates their parents to discover how to do things for the first time. They should maintain discipline and not break from the structure, repeating each action in accordance with their instruction. This will provide the foundations for their success. At this stage, they are merely *doing Agile*. In time the student will move to *Ha*, (think *aha*!) and *detach* from the structures they were taught as they begin to innovate. Now they are *thinking Agile*. Eventually, the student may ascend to *Ri*, where they completely *transcend* the forms and open the door to creative technique. They have moved beyond the lessons their teachers taught them, and have now gained mastery. However, they still do not violate any universal laws inherent in their discipline. Instead, everything they do has become an embodiment of agile, which now resonates from the core of their being. Every action emanates from this core identity. Agile is now who they are. Like Jonathan Livingston Seagull, who "was not bone and feather, but a perfect idea."

Non-Being

In contrast to the concept of being is non-being. This can include, but is not limited to, the fear of death. However, non-being doesn't just hint at the idea that our lives will one day come to an end, it also speaks to the void, the feeling of emptiness that may creep, and sometimes pervade, our lives. Also, it can refer to that period *before* our birth. This idea was expressed best by Charles Pierce in The Logic of Events:

> *We start, then, with nothing, pure zero. But this is not the nothing of negation. For not means other than, and other is merely a synonym of the ordinal numeral second. As such it implies a first; while the present pure zero is prior to every first. The nothing of negation is the nothing of death, which comes second to, or after, everything. But this pure zero is the nothing of not having been born. There is no individual thing, no compulsion, outward nor inward, no law. It is the germinal nothing, in which the whole*

> *universe is involved or foreshadowed. As such, it is absolutely undefined and unlimited possibility – boundless possibility. There is no compulsion and no law. It is boundless freedom.*

The message here is simple, but powerful: anxiety is unavoidable. Irvin Yalom felt that in order to live an authentic life, one must face the reality of death and limitation. By the very nature of our lives, and the inevitability of our deaths, we experience an undercurrent of anxiety, often manifested as guilt. It's the guilt we feel when we leave work early to watch our child's basketball game, or when we stay after work late and miss having dinner with our loved ones. There are competing demands on our time, and thus every decision becomes a value proposition. In the end, we often sacrifice our authentic selves for the things we are told to value. On their deathbed, how many have looked back and concluded they should have spent more time at the office?

Self vs. World

The distinction between being and non-being parallels another central theme in existentialism, that of self vs. world, or rather, our individual relationships to the world we're surrounded by. In The Essential, Ken Wilber expanded this line of thought:

> *In any developmental sequence, what is whole at one stage becomes merely part of a larger whole at the next stage. A letter is part of a whole word, which is part of a whole sentence, which is part of a whole paragraph, and so on. Arthur Koestler coined the term "holon" to refer to that which, being whole in one context, is a part of a wider whole in another. With reference to the phrase "the bark of a dog," for example, the word "bark" is whole with reference to its individual letters, but a part with reference to the phrase itself. And the whole (or the context) can determine the meaning and function of a part – the meaning of "bark" is different in the phrases "the bark of a dog" and "the bark of a tree." The whole, in other words, is more than the sum of its parts, and that whole can influence and determine, in many cases, the function of its parts.*

A person is complete and unique unto themselves, yet they also comprise part of a greater, contextual whole: that of society. Individuals exist as a part of a family, an organization, a community, a culture; which lends depth, context, and meaning to their lives. Conflicts arise when our individual desires fail to align with the external desires that are placed upon us.

In society, the collective is usually afforded the moral-right over the individual, a concept referred to in philosophy as utilitarianism. Collectivism inevitably brings up the concept of *deindividuation*. The term itself arose from Pepitone and Newcomb in 1952, and has been defined as, "In groups or crowds, the loss of awareness of one's own individuality."

Aldous Huxley famously portrayed a society entirely built on the idea of the collective in his book Brave New World, which replaced the idea God with a new idol to worship: Henry Ford. Huxley uses Ford, inventor of the assembly line, to represent the collectivist nature

of Industrial society. In Huxley's Brave New World, the unique and creative nature of the individual has been ostracized, so that only those who conform to the strict standards of modern society remain.

Agile seeks to displace this dystopian view of the conflict between the individual and the collective, commonly seen as an extension of industrialization, with a new paradigm for managing employees and the work they do. In this new paradigm, the purpose-motive is moored to the profit-motive. We use strengths-based approaches to align individual talents and interests with the work they perform. Knowledge workers are entrusted and empowered. They are given autonomy, the chance at mastery, and a sense of purpose. Otherwise, what has it all been for?

Finding Meaning

Non-being represents the acknowledgement and acceptance that one day we, too, shall die. *Being* emphasizes authenticity in our relationships, i.e., being true to ourselves (the individual) and in our relationships with others (the collective). Living an authentic life gives *meaning* to the limited time we have.

Freud called the sense of immediacy, and consequential anxiety, caused by the knowledge that all life is finite, the "Death Instinct." Erich Fromm reaffirmed that notion in the following sentiment:

> *Being aware of himself, [Man] realizes his powerlessness and the limitations of his existence. He visualizes his own end: death. Never is [Man] free from the dichotomy of his existence."*

Death is a *process*, even though people often wish to view it as an event. The *moment* of death is an event, but death itself is part of the same process we call *life*. In fact, they are not simply a *part* of the same process; they *are* the process together, simultaneously. Because the majority of humanity views life and death as two separate processes, it coincides that they *fear* death while *embracing* life. This is the dichotomy that Erich Fromm was talking about, and it is through this dichotomy that anxiety arises.

Existentialists reason that it is the inevitability of death that gives meaning, purpose, and direction to our lives. Death adds the contrast from which we are able not only to *define* our existence, but search for meaning *within* that existence. If life were static, so too, would be our efforts. There would be no need to exercise, to feed the mind, to nourish the soul. Death gives consequence to our actions and meaning to the *choices* we make. Therefore, death emphasizes both our freedom to choose and the meaning we attribute to those choices.

We are free to operate within the confines of our mental and physical limits, as defined by genetic inheritance and the laws of physics. Fate does not exist until after the fact. You see, we are free to do anything we choose, but after the choice has been enacted, we cannot change the past. Fate is *post facto*. We are left with the effects of our cause. In order to

become, as Maslow described, a self-actualized individual, we must assume responsibility for the choices we make.

People want their lives to have meaning. They want for the work they do to matter. The individual derives meaning based, in part, on what they contribute to the collective. This is particularly true for millennials. People want to be remembered not for how they died, but how they lived. *Memento mori.* What is a life worth, and how is it measured?

The autonomy that Agile creates provides freedom and responsibility in equal measure. It provides a chance to reach the pinnacle of Maslow's hierarchy of needs, because it's about our freedom to choose a course of action, and then to accept responsibility for that choice. We must first become self-directed before we can become self-actualizing.

Agile creates a high level of transparency, and that sets the stage for employees to develop authentic relationships. The work place is a naturally competitive environment, but Agile shifts this dynamic and puts everyone on the same team, equally responsible and accountable for delivering *all* the work, not just their portion of it. Everyone can see the value that everyone else brings.

Existentialism can provide a powerful frame-of-reference for understanding and sharing in the humanity of others. Maintaining a phenomenological perspective will help people develop empathy, genuineness, and unconditional positive regard. These are the hallmarks of a person-centered approach. This will enrich social interactions and build a foundation of trust. Understanding how people interpret the world around them, and what they connect with, will help you connect with them in a meaningful way. People are the most valuable asset a company has. Don’t you think it’s time we began treating them as such?

CO-LOCATION

Servant leadership entails more than just recognizing the self in others and then supporting their efforts. It's also about creating an environment where they can be successful. Not only does this involve the tools they use and the architectural platforms they develop on, but it also concerns the team and organizational structures, as well as the space design that inhabits them.

Co-location and open space designs are two places where I commonly see organization's fall short. For starters, I often hear the following reasons why team members can't be co-located: "We hire the best people for our projects regardless of where they live." "We have quota numbers to meet for a certain percentage of our personnel to be located offshore, because of the cost advantages." There are, of course, ways to deal with this. But there is a denial here as well, around the hidden costs of inefficient team structures. People assume, albeit incorrectly, that a co-located model is more expensive and less productive. They don't stop to consider the compounded costs of all these hand-offs and inefficiencies over time, from elongated feedback loops to poor communication.

Furthermore, for those organizations that do co-locate their Agile teams, they often introduce additional inefficiencies that are predominantly a result of how they've implemented their space designs. The practice of co-locating teams using an open-space floor plan is a common technique to increase collaboration and collapse the feedback loop among Agile team members. This works, so long as the work the team is focused on requires a high level of collaboration. If not, then the environment can be counterproductive, due to the increased level of distraction resulting from interruptions and background noise. It is prohibitive to have teams co-located together, if the work they're doing is disparate. Instead, team members should be co-located with members of their *own* team, but separated from *other* teams, to ensure the osmotic communications they are exposed to are relevant to the effort at hand. There are a few key concepts and considerations to keep in mind when moving to an open space design. The first concept we will look at is team size.

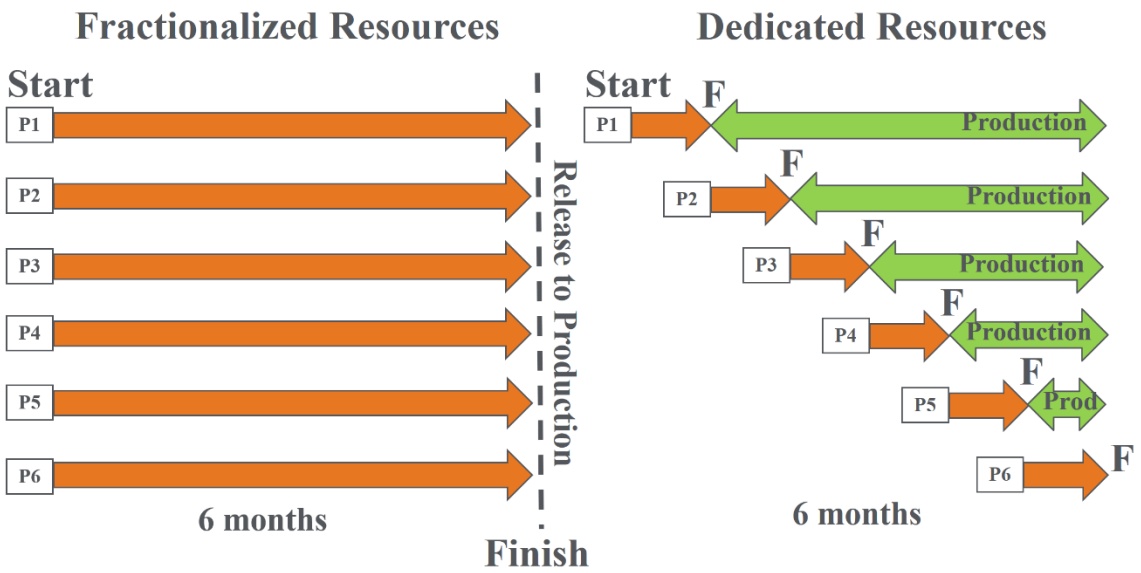
Team Size

Ideal team size is often denoted as 7 plus or minus 2 (i.e., 5 to 9 people per team), though teams of 3 to 9 have proven effective. I can't tell you the number of times I've seen large Agile teams and been told they couldn't be made any smaller. This is usually the result of fractionalized resources, over-specialization, poor architecture, or heavily dependent environments. Team size has a direct correlation to individual productivity levels, so a large team is – quite simply – an inefficient team. There has been a considerable amount of research done on ideal team sizes, and the evidence is clear: as team size goes up, productivity goes down. Here are some of the reasons:

- **Miller's Law**
    - The key number is 7, plus or minus 2: this has been shown to be the limit on our ability to process and retain chunks of information. Modern research suggests the number is actually closer to four.
- **Richard Hackman**

  - Harvard professor whose broad research shows 4.6 members as the average number for ideal team size, based on the greatest levels of productivity.
- **QSM**
  - Thorough research that shows that variation goes down and productivity goes up as team sizes reduce.
- **Brook's Law**
  - Adding more people to a late project only makes it later (i.e., "the mythical man month").
- **Dunbar's Number**
  - A person only maintains 150 social (not collaborative) relationships.
- **Conway's Law**
  - Teams produce designs that mirror the communication structures of their organizations (e.g., distributed teams create modular products).

As it pertains to fractionalizing or dedicating personnel to a single team (and thereby a single project), here's a simple example to show not just the increase in productivity levels that you'll gain by doing so, but also the actual ROI benefits associated with it:



6 projects, over 6 months, will produce (5+4+3+2+1) 15 months of added ROI!

In the left column above, we see a traditional example where a "resource" may spend their time working on six different projects simultaneously. If each project lasted one month in duration, then at the end of six months, six projects would be delivered. However, if the "resource" worked on only one project at a time, prioritized by business value, then they would deliver the first project at the end of the first month. This would provide five months of additional ROI (i.e., value to the business), who then gets to consume the service in an accelerated timeframe. This process would continue, with each project in this example taking one month to complete, and all projects thereby providing additional ROI back to

the business with the exception of project six, which would be delivered at the end of the sixth month. 6 projects, over 6 months, produces 15 months of added ROI. Not only are dedicated resources better for the teams, but it's also good for the business!

It goes beyond just time-to-completion, however. There's actually an efficiency loss that occurs when you fractionalize your resources, which reduces the overall amount of time that resource will spend doing actual product-focused work:

Simultaneous Projects	Percent Allocated	Efficiency Loss
1	100%	0%
2	40%	20%
3	20%	40%
4	10%	60%
5	5%	75%

If you are working on 5 projects, then the research shows that – on average – you are only spending about 5% of your time working on each of those efforts, for a total of 25% of your total capacity. The rest of that time (75%) has been lost to the inefficiencies inherent in the myth of multi-tasking. This is due, in large part, to context switching, which I'll explain more about in a little while.

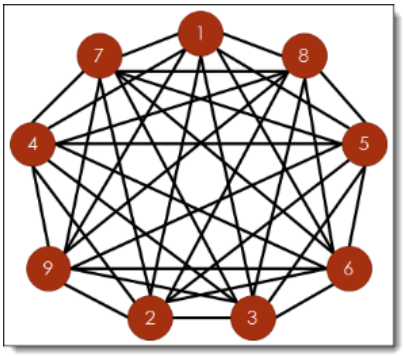
Communication Channels

Team size is also a determining factor for effective communications. This is because the number of people on the team will ultimately determine the number of communication channels, which will also influence how much time people spend to maintain effective communication across all those channels. Understanding the impact of this is essential.

The communication channel formula is: N(N-1)/2

Thus, the communication web grows exponentially, and becomes more entangled, as additional team members are added:

- 1(1-1)/2 = 0
- 2(2-1)/2 = 1
- 3(3-1)/2 = 3
- 4(4-1)/2 = 6
- 5(5-1)/2 = 10
- 6(6-1)/2 = 15
- 7(7-1)/2 = 21
- 8(8-1)/2 = 28
- 9(9-1)/2 = 36



Research has shown that communication breakdowns inevitably start to occur as the number of team members expands. As it so happens, nine team members (which equals 36 communication channels) is the first critical point where communication breakdowns begin to occur (i.e., it is simply not possible for people to effectively communicate across 36 separate channels in a meaningful way). Thus teams actually become less efficient as team size grows. Large team sizes therefore become the antithesis to a lean, agile approach. As team sizes get larger, the left hand no longer knows what the right hand is doing. While two heads may be better than one, 10 heads are not able to synchronize their efforts.

Collaborative Spaces

DuPont pioneered the open, collaborative space design during the 1950s. After experimenting with this open office concept for some time, they chose to abandon it, having noted no verifiable increase in productivity. Since then, other businesses have adopted the model, and it has become particularly common in the software development industry.

The large body of research that has since been conducted on the topic has generally been mixed, though recent studies have concluded that the open space design results in:

- A decrease in employee satisfaction (i.e., morale)
- A decrease in productivity

The research article "Traditional Versus Open Office Design" (Brennan, Chugh, Kline, 2002) took the following factors into consideration:

- **Physical environment**
  - Amount of storage space, work surface area
- **Physical stressors**
  - Lighting levels, noise
- **Team member relations**
  - Inclusion, approachability
- **Performance**
  - Ability to focus, ability to stay on task
- **Protocols**
  - Had office protocols been established and were they followed?

This particular study concluded that employees were significantly less satisfied with the physical environment of the open office design, and that their dissatisfaction remained constant over time (i.e., they did not grow to embrace the change). The results were similar for all additional measures.

However, another research article entitled "Employee Reactions to Office Redesign" (McElroy and Morrow, 2010) surmised that "office redesign is an effective strategy for implementing organizational change." This article looked at organizational behavior and methods for inducing behavior modification. It's important to understand the following: space = culture. Behavior is influenced and determined through space design.

An adaptable business model requires an adaptable space. According to Stewart Brand,

author of the seminal architectural book How Buildings Learn (1995), "Age plus adaptivity is what makes a building come to be loved. The building learns from its occupants, and they learn from it." Christopher Alexander, professor of architecture at the University of California, Berkeley, offers a similar point of view: "What does it take to make comfortable little modifications in a way that once you've made them, they feel integral with the nature and structure of what is already there? You want to be able to mess around with it and progressively change it to bring it into an adapted state."

Open space design offers increased flexibility (i.e., adaptability) and an effective strategy for implementing organizational change. However, it generally results in a decrease in employee morale, and productivity, and that decrease is sustained over time. There are some important factors to consider if efficiency is a key driver. While open space designs generally drive some positive behaviors, such as transparency, collaboration, responsiveness, adaptability, and collective ownership, they also introduce a myriad of distractions. A prominent feature of co-located spaces is the concept of osmotic communication. This is defined as "Communication which occurs as a result of people sitting in the environment. One team member overhearing two other team members conversing in the war room and thus becoming informed would be an example of osmotic communication." (Crowe, 2012)

Osmotic communication can be extremely beneficial, allowing a team member to join in the discussion when it relates to one of his or her knowledge areas or project interests. However, osmotic communication can also result in detrimental noise when it does not have any relevance to the team member. For this reason, many modern open-space advocates recommend co-locating the team, while simultaneously keeping them separate from other teams. Thus, you want team members that work closely together to be near one another, but you want separation from other teams and from other team members whose work does not closely align with their own.

The research article "Stress and Open-Office Noise" (Evans and Johnson, 2000) draws the conclusion that "open-office noise elevated workers' urinary epinephrine levels, but not their norepinephrine or cortisol levels, and it produced behavioral aftereffects (fewer attempts at unsolvable puzzles) indicative of motivational deficits. Participants were also less likely to make ergonomic, postural adjustments in their computer work station while working under noisy, relative to quiet, conditions. Postural invariance is a risk factor for musculoskeletal disorder." In the results section of the article, it is noted that workers in both the control group (traditional office space design) and experiment group (open office space design) reported feeling similar levels of stress at the end of a three-hour period, but those in the open office space had additional physiological signs that indicated elevated stress levels above that of their counterparts in the control group. The stress of the environment had a direct, measurable impact on them physically.

Context Switching

The optimal number of tasks for each team member is no more than two in progress at a time (Clark and Wheelwright, 1993), which allows someone to switch over to another task if the first one should become blocked. As the number of tasks increases beyond two, the

amount of time it takes to context switch is inversely proportionate to their level of productivity.

Thus, in an open space design, particularly with large team sizes, the amount of context switching that takes place due to interruptions (either from members of their own team or from other teams) results in a significant decrease in productivity. If the intent is to increase efficiency, then it will be necessary to limit these context switching disruptions, and thereby limit the number of concurrently assigned tasks each team member has. The more tasks a team member has, and the more projects they're involved in, the higher the rate of disruption (since more people will require their expertise). Large teams mean more disruptions (after all, you've got more communication channels to maintain!

Self-Organizing Behavior

Space design should grow organically. For a flexible, adaptive space that also promotes a shift in culture (i.e., induces changes in behavior), the focus should be on team-specific needs. What may appear efficient for management may not be ideal for the team, and vice versa. The team should be the primary concern: who they interact with, how they work together, and in what capacity. The goal is to promote self-organizing behavior, to let the teams reorganize to build in their own efficient work flows. This promotes collective ownership. It also contributes to producing a "culture of innovation." The whitepaper "Creating a Culture of Innovation" (Krieger, 2010) states the following:

> *Since the Industrial Revolution, there have been three main business innovations. The first was neoclassical economics to drive a supply-and-demand economy. The second occurred in the 1980s and 1990s when companies like Toyota, GE, and others used process improvement (Six Sigma, lean manufacturing) to drive profitability and growth.*
>
> *In today's marketplace, these models no longer provide the competitive advantage they once did. So the third major business model will be the driver of innovation moving forward. This model is behavioral economics.*
>
> *Whether these three models were product driven, knowledge driven, or innovation driven, one thing remains constant: They all need people and managers. To execute, you must choose to use people as allies rather than adversaries. Behavioral economics is the science that allows this to occur. There are three mechanisms or levers that an organization can pull to drive innovation using behavioral economic principles. They are culture measurements, capability assessments, and selection of the idea catalysts (the organization's people).*

An organization's most valuable asset is its people. Create an environment that will attract the live hearts and the live minds. Space design isn't just about making people more efficient; it's also about attracting and retaining high-quality talent, by creating an environment conducive to the way they want to work. The environment should be collaborative, with autonomy and accountability, in equal measure.

The design should grow bottom-up, not top-down. To quote Abraham Lincoln, "by the people, for the people." Avoid the pitfalls of open space design that can produce a reduction in both productivity and morale. Create an adaptable space. To drive efficiency using space design (which should augment the company's overall process), focus on smaller team sizes and at least some team segregation. "Mega pods," or "mega teams," have not proven to be an effective method for increasing productivity.

In the research article "Individual Flexibility and Spatiality" (Värlander, 2011), the conclusion states, "This article sheds light on how space shapes and is being shaped by its incumbents." In addition, "for practitioners, the findings of this study underscore that an awareness of the unintended and emergent consequences of spatial design is imperative in order not to have naïve expectations on the influence of spatial layouts. Spatial design can be an efficient tool in implementing change, but there is a need for awareness of the unpredictability of spatial design, and simplistic views of openness as unequivocally leading to flexibility, innovation, and other favorable or desirable organizational outcomes need to be challenged."

There needs to be an initial design that will promote adoption (i.e., team members to buy-in to the open design concept) so that collective ownership becomes an ingrained principle, and people begin to self-organize and adapt to their surroundings rather than protest them. This is what will drive efficiency in a sustainable, innovative way.

OPEN SPACE DESIGN

Space design is a reflection of the values of the organization, and it will drive certain behaviors. If you want an open, transparent, and collaborative culture, then the design of your spaces should emulate that goal.

The Six Ss

In his seminal architectural design book, How Buildings Learn: What Happens After They're Built (1995), Steward Brand describes how buildings and space evolve over time to meet the needs of their inhabitants, through small feedback-loop adaptations. When reshaping work environments, Brand indicates there are six elements to consider. They are collectively known as the *Six Ss*:

- **Site**
  - Location
  - The rate of change is extremely low
- **Structure**
  - The foundation and load-bearing elements
  - The rate of change is 30 to 60 years
- **Skin**
  - Exterior surfaces
  - The rate of change is 20 years
- **Services**
  - Electrical/communications wiring, plumbing, sprinkler system, HVAC
  - The rate of change is 7 to 15 years
- **Space plan**
  - Interior layout – where walls, ceilings, floors, and doors go
  - The rate of change is every 3 years or so
- **Stuff**
  - Furniture
  - The rate of change is daily to monthly

Palettes, Postures, and Quadrants

In an open office space design, use different palettes (i.e., colors) and postures (i.e., furniture of various types adjusted at different heights) to create a space that conforms to a variety of individual preferences. Design work spaces with the following *four quadrants* in mind:

- **I/Shared**
  - Private temporary personal space
  - Open temporary personal space
- **I/Owned**
  - Individual workspace
  - Semi-private permanent space
- **We/Shared**
  - Private collaboration
  - Open collaboration

- **We/Owned**
  - Team workspace

This is similar to the "caves and commons" approach in Agile environments, but it meets a broader number of employee needs. For example, an "I/Shared" workspace might be a private office space that is available on a first-come, first-served basis or a comfortable chair in an open-space environment that provides some solitude away from the team. Meanwhile, an "I/Owned" workspace might consist of the individual's desk, file cabinet, and chair. A "We/Shared" area might constitute a nearby conference room that is available to all teams or a collaboration area available within the open space. In contrast, the "We/Owned" workspace represents the co-located environment in which the team resides on a daily basis, usually made up of individual workspaces (i.e., "I/Owned") arranged to form a pod (with a table in the center for collaboration) or formed to create the appearance of a large table out of the jointly arranged desks.

Universal Design Principles

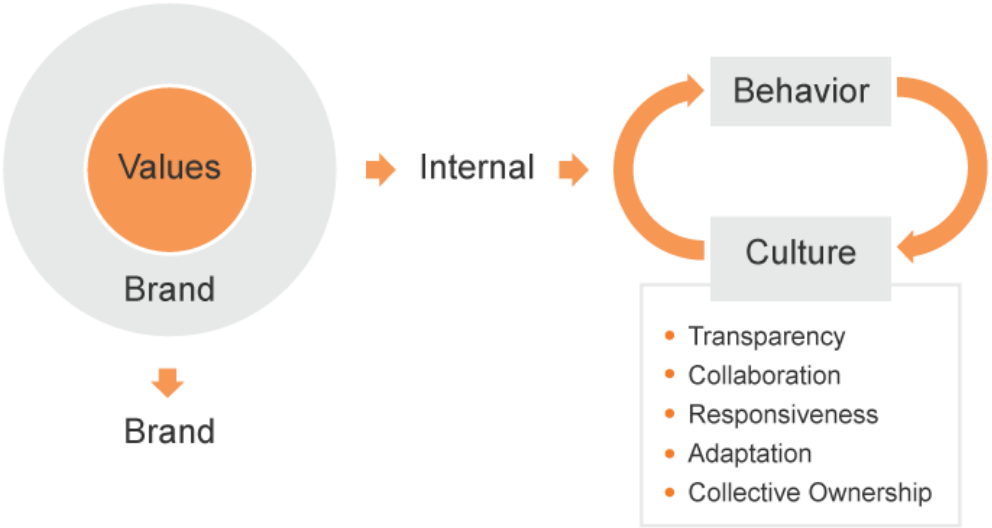
Whenever possible, incorporate universal design principles into the space redesign effort. With this method, you design to the widest margin of the population. It inherently accommodates the greatest number of people, regardless of what life challenges they may have to contend with (for instance, the use of levers instead of doorknobs, to make access easier for those with disabilities).

Form Follows Funding

As Stewart Brand says in his book, form seldom follows function… instead, it follows funding. Organizations have financial constraints. Therefore, rather than basing space designs on ideal working conditions, they base them on the level of funding allowed for modification to these environments. The higher up you go in the Six Ss, the higher the price tag for the redesign effort. For instance, If you have to change the location of services (such as electrical), the cost increases dramatically – especially if this requires extensive changes to the structure of the building. For instance, suppose the electrical system currently runs through a concrete floor, and there is no room to insert a floating floor on top of the concrete to accommodate the desired design. In such a case you will either make extensive changes to the concrete floor itself or route the electrical power through the ceiling (which may not be an option). At any rate, these dependency changes inflate the price tag significantly. For this reason, you'll want to remain cognizant of the Six Ss, and what your goals are for the space redesign effort.

Space Equals Culture

According to David Lathrop, the Director of Research and Strategy at Steelcase: "Space equals culture. Our values define our brand, both internally and externally, and space design should be a reflection of our values." If the organization places value on openness and transparency, then the space design should reflect those values and serve to augment that type of behavior.


Values
Brand
Brand
Internal
Behavior
Culture
Transparency
Collaboration
Responsiveness
Adaptation
Collective Ownership

HOW TO GET STARTED

In a large organization, it can often take several years to transition to agile. This is a big change in culture, and change is hard work. Not only does the organization need to adopt new practices, but it needs to value new principles. This change in philosophy will be in stark contrast to the old command-and-control style approach that likely preceded it. Software development organizations are moving away from traditional, Waterfall based project management methodologies, to Agile frameworks. These organizations are seeking a competitive advantage by removing process heavy techniques that rely on obstructionist change management approval processes, and instead are adopting a more collaborative and adaptable approach.

Making the transition from a traditional Waterfall-based methodology to an Agile framework will require a significant change effort. To effectively manage the adoption process, and then sustain the change, it will be beneficial to use a formal change model. The organization should develop a change strategy, and then measure their progress against the plan. Like the empirical process control being implemented, progress should be monitored against the plan, and the plan should be adapted based on feedback and/or in response to unintended consequences.

The Change Process

The process of change management includes the following steps (Hayes, 2010):

1. **Recognize need and start change process**
2. **Diagnosis**
    a. Review the present state
    b. Identify the future state
    c. Quality of the vision
3. **Plan and prepare to change**
4. **Implement the change**
5. **Sustain the change**

In addition, throughout the entirety of the change effort, the following ongoing activities will need to take place:

- **Review progress and feedback**
- **Manage the people issues**
    - Power, politics, and stakeholder management
    - Leadership
    - Communication
    - Motivating others to change
    - Support for others to help them manage their personal transitions
    - The change manager's mode of intervening and the effect this has on the change relationship

Change Models

Lewin (1951) suggested a three-step model for change that included the following stages:

1. **Unfreezing**
2. **Moving**
3. **Refreezing**

During *unfreezing*, driving forces should be emphasized, and restraining forces should be weakened. This will destabilize the current environment, and ready the change. It should prompt people and teams throughout the organization to begin letting go of old behaviors, and seek out new and more effective alternatives.

When *moving*, the driving and restraining forces are adjusted to the point that the organization shifts to a new level. This can be accomplished by modifying attitudes, beliefs, processes, systems and structures. Smaller changes may only adjust a subset of these attributes, while larger changes will require that all five be modified.

Once the organization has shifted to a new level, *refreezing* occurs. This allows the change to be reviewed, progress measured, and the ongoing effort fine-tuned. In addition, the change is reinforced, and regression can be avoided. Organizations often perform the first two steps in Lewin's change model, but if they exist in a turbulent environment, will forego the third step. This is a dangerous scenario, because refreezing allows the change effort to be sustained. Without this crucial step, employee behaviors may revert back to old roles and old habits.

Like the Agile process itself, which is based on the kaizen principle of "good change", Lewin's three-step model uses an iterative and incremental approach to transition and sustain the effort. The organization should make small, progressive changes, and then freeze the change effort to allow for adaptation based on review, and to embed the new practices. This method is consistent with the gradualist paradigm, which posits that "fundamental change (organizational transformation) can occur through a process of continual adjustment, and does not require some major discontinuous jolt to the system in order to trigger a short episode of revolutionary change. According to the gradualist paradigm, change is evolving and cumulative." (Hayes, 2010).

However, the punctuated equilibrium paradigm acknowledges that incremental changes occur when the focus is on continual process improvement, during periods where the industry is in equilibrium. In this scenario, small adaptations are implemented to gain a competitive advantage. Meanwhile, a fundamental change at the deepest levels of the organization cannot be sustained through incremental efforts. Incremental changes should be used for optimization, while transformational (i.e., revolutionary) change will require breaking with the past, and forming new patterns. The focus in transformational change is to do things differently, rather than doing them better. Sometimes, it may even include the decision to do different things. This is because transformational changes occur during periods of disequilibrium, and may call into question the very purpose of the organization. (Hayes, 2010).

Once you've made the shift, the gradualist paradigm will work well for continual process improvement. However, when an organization considers switching to Agile, this is often

during a period of disequilibrium, and thereby will initially require transformational change, which should then be followed-up with small, incremental optimization efforts.

Meanwhile, open systems theory focuses on the interrelated components between the organization, and the larger system it is embedded in (i.e., environmental conditions) that are causing the need for adaptation. It is a quest for internal and external alignment, with no specific end state, but rather a focus on continual optimization. Agile is closely aligned with open systems theory, since they share the following characteristics (Hayes, 2010):

1. **Embedded within a larger system**
2. **Able to avoid entropy**
3. **Regulated by feedback**
4. **Subject to equifinality**
5. **Cyclical in their mode of functioning**
6. **Equilibrium seeking**
7. **Bounded**

Scrum, for example, is embedded in a larger system that requires customer engagement, alignment with organizational directives, implementation of regulatory requirements, and adaptation to emerging industry trends. It is able to avoid entropy through adaptive planning techniques, and customer collaboration. Scrum uses product and process feedback loops (e.g., sprint review and sprint retrospective ceremonies) to review outputs and then regulate inputs for the next iteration. Scrum is subject to equifinality, since it emphasizes continual process improvement, and thereby adapts process techniques to the unique circumstances of each team. Scrum uses sprints, otherwise known as iterations, which are cyclical in their mode of functioning, and use a repetitive cycle of input, throughput, and output. The Scrum framework seeks equilibrium through patterned events and a sustained pace. Last of all, Scrum is bounded by limiting the amount of work-in-process (WIP) at any one time.

Realignment

George Schlitz (2013) described five facets of an organization that may need altered when switching to an Agile framework. These include:

- **Execution**
  - Skills, practices for individuals and teams
- **Delivery**
  - Product development and delivery
- **Product and Business Model**
  - Product management and strategy
- **Organization**
  - Structures, processes and systems by which work gets done and is organized
  - Collective beliefs, perspectives and habits by which people make sense of things
- **Leadership**
  - Leadership and management styles and beliefs about what constitutes effective leadership and management
  - Orienting vision and environmental design

Schlitz recommends that each of these facets be considered and managed throughout the change effort. The current state of the organization should be defined, and the future state identified, with the necessary changes being made across all five facets using a build-measure-learn feedback loop, which closely resembles Lewin's "unfreeze-move-refreeze" approach.

According to Mike Cohn (2010), if the implications of becoming Agile are not transferred into other parts of the organization, then *organizational gravity* will kick in, thereby reverting the change effort back to the previous state of existence. In his book, Succeeding with Agile: Software Development Using Scrum, Mike states the following:

> *"Organizations and organisms evolve to fit their environments. According to the principle of selection, those traits most likely to help an individual or group survive in the organization will be the ones retained. It is the organization's leaders and managers who define what traits help groups or individuals survive. If Agile values such as openness and transparency lead to survival in the form of promotions and public praise, those behaviors will be the ones individuals select."*

Mike describes four components that may require realignment under agile:

- **Human Resources**
- **Facilities**
- **Marketing**
- **Finance**

For instance, human resources should consider modifying personnel evaluations so that they emphasize team behaviors, rather than individual performance. By shifting to team-oriented criteria, the organizational environment will emphasize the values inherent in Agile frameworks, and thereby promote individual adoption of the desired behaviors. Many organizations are uneasy about switching to a group accountability model, since they have an intrinsic preference for the traditional, individual approach. (Katzenbach and Smith, 1993).

Similarly, frameworks like Scrum use co-location to emphasize collaboration and collapse the feedback loop. Consequently, facilities may need remodeled in order to create an open, collaborative environment that will be conducive to this new approach. The space design should augment the team, and reinforce the values and behaviors of the change effort.

Adopting agile principles requires a shift in focus for Finance, to an adaptable model that welcomes changing requirements. The emphasis is placed on customer collaboration, rather than contract negotiation. Consequently, well-defined contracts that prohibit change should be limited in use, opting instead for generalized verbiage that allows the two parties to negotiate throughout the development effort. This is often accomplished through timeboxing the activity, while leaving scope flexible, and then collaborating with the customer to ensure the most important items are being completed first.

Making the Transition

Unlike other change efforts, Agile adoption can be particularly challenging, because it requires implementation from the top-down and bottom-up, simultaneously. Agile is about team empowerment, so top-down enforcement is not an ideal scenario. Instead, leadership needs to demonstrate an organizational commitment to Agile, and prompt the change effort, but afterwards they have to empower the teams and let the change emerge throughout the organization.

In alignment with the steps of the change process outlined by Hayes, the first step should be well underway at this point: the organization has *recognized the need* (or desire) to switch to an Agile framework, and is initiating the change process. This should prompt the next step, *diagnosis*, during which the present state is reviewed, a desired future state is defined, and the quality of the vision is clarified.

During the diagnosis phase, tools and assessments should be used to gain organizational insight. Both SWOT and Force Field Analysis are good tools to use early on, because they are quick, collaborative, and insightful. Once a foundation has been acquired with these rudimentary tools, then it can be progressively elaborated into more complex models.

Next, the organization must *plan and prepare to change*. Using the information obtained during diagnosis, organizational barriers to the change effort should be targeted for removal. This is akin to Lewin's *unfreezing* stage, during which the drivers are emphasized, and the resistors weakened.

With the ground readied, the organization should then proceed to *implement the change*. This will be the *moving* stage of Lewin's three-step model. The organization should begin with a pilot effort that will combine projects, and team members, that are ideally suited for the change, and can then serve as influencers for the overall adoption effort. The pilot team should continue to optimize and adapt course, as necessary, to improve performance. In Scrum, the team should be self-organizing, and consequently, self-regulating. Retrospective ceremonies should be held at the end of each sprint, so the team can reflect on their previous effort, and continue to fine-tune process efforts.

According to the whitepaper, A CIO's Playbook for Adopting the Scrum Method of Achieving Software Agility, a gradual adoption process is well-suited for rolling Scrum out in larger organizations. Initial efforts should be focused towards: "assessing the organizations readiness for agility; providing initial training for the early participants; and building the Product Backlog for the initial projects." A first pilot project should be identified. This would "demonstrate the positive benefits of improved software agility within the organization", thereby increasing interest and excitement for others to try out the new process. After a few successful pilot projects have completed, it would then be time to "expand the usage of Scrum and its benefits to a significant subset of the development organization." (Schwaber, Leffingwell, & Smits, 2005).

To *sustain the change*, leadership should also focus on tracking and removing organizational barriers that are a hindrance to team performance. This uses the Agile

principle of continuous improvement, and applies it to the organizational transition effort itself. To do this, a list of organizational impediments should be identified and tracked (this can be done through a backlog), with the goal of removing anything that obstructs the process. In Lean, this is known as value-stream-mapping, and then subsequently removing identified waste. Thus, organizational change efforts should be temporarily *refrozen*, reviewed, and retargeted.

As pilot projects are met with success, the organization should begin to expand their efforts. A seed approach is helpful here (Cohn, 2010), where a member from a previous team will join a new team and provide them with process knowledge transfer. The organization should continue to loop through Lewin's three-step model of unfreezing, moving, and refreezing; all while expanding their efforts and adapting to unforeseen consequences.

Managing the People Issues

When confronted with "too much change in too short a period of time" people may become disoriented and experience future shock (Toffler, 1970). According to Mike Cohn (2010), many organizations and employees "have been suffering from future shock for years." He cites the following causes:

> *"Teams are asked to do more with fewer people. Outsourcing and distributed teams have become increasingly common. These adjustments were preceded by the rush to move applications to a client/server model, then onto the web, and then into services. Add to these the constant, and constantly accelerating, rate of change in technology itself – new languages, new tools, new platforms – and future shock is now. It should not be surprising that transitioning to Scrum can often be the change that pushes people into future shock. The pervasive nature of adopting Scrum and the fundamental changes it causes in how people work and interact have a higher risk of triggering the future shock effect."*

An external change agent, such as an Agile Coach, could be brought into the organization to provide an objective point-of-view. They would act in a consulting role, offering organizational advice, and facilitating the process. However, the change initiative would have to belong to the business, and it would have to be aligned with the organization's overall strategic vision, culture, and desire. The change must derive from within (i.e., internally, rather than externally). Consequently, leadership must become the champions of this effort, and advocate on behalf of it. While some teams will be excited to transition, and may event start using Agile practices on their own, for the entire organization to transition, and then sustain the change, the effort must be reinforced by leadership.

The CIO must serve in the role of ScrumMaster for organizational change. According to Ken Schwaber (2005), the differentiator between success and failure for top-down implementations of Scrum is the CIO. Therefore, it will be imperative for the CIO to be heavily involved in the transition effort: advocating the process, removing organizational impediments, and acting as a change agent (rather than just a change catalyst – i.e.,

someone who initiates the change, but then no longer takes an active role to ensure its long-term success).

Leonard Sayles (1964) points out the contradictory nature of the leadership role in an open systems model:

> *"The one enduring objective is the effort to build and maintain a predictable, reciprocating system of relationships, the behavioral patterns of which stay within reasonable physical limits. But this is seeking a moving equilibrium, since the parameters of the system (the division of labor and the controls) are evolving and changing. Thus, the manager endeavors to introduce regularity in a world that will never allow him to achieve the ideal."*

Sayles has correctly identified that environmental conditions (e.g., market demands, regulatory requirements, pace of technological innovation, etc.) continue to evolve, and thus the focus should be on continual process improvement, rather than towards achieving a singular, static, and complacent state of existence. This is precisely why the servant leader approach of Agile is so effective… because the leader must serve the team, rather than vice versa. The team, and thereby the organization, should be in a state of perpetual adaptation.

In what is equally good advice for both people, and organizations; the following quote from Erich Fromm is an excellent reminder about the need to grow, and adapt, in order to remain viable:

> *"We must be in order to become, and we must become in order to be."*

References

Alexander, C. (1964). *Notes on the synthesis of form*. London: Oxford University Press.

Aristotle. (2003). *The giants of philosophy*. Knowledge Products.

Ayres, T. (2004). *That's not in my american history book: A compilation of little-known events and forgotten heroes*. Taylor Trade Publishing.

Brand, S. (1994). *How buildings learn: What happens after they're built*. United States: Viking Penguin.

Brennan, Chugh, Kline. *Traditional versus open office design*. Retrieved October 8, 2012, from http://search.ebscohost.com/.

Cohn, M. (2010). *Succeeding with agile: Software development using Scrum*. Boston, MA: Addison-Wesley Signature Series.

Crowe, A. (2012). *The PMI-ACP exam: How to pass on your first try*. United States: Velociteach.

Dass, B. R. (1997). *Be here now*. Tennessee: Kingsport Press.

Evans, J. *Stress and open-office noise*. Retrieved October 8, 2012, from http://search.ebscohost.com/

Fromm, E. (1976). *To have or to be?* New York: Continuum.

Funk, R. (1999). *The Erich Fromm reader*. New York: Humanity Books.

Guidice, D. L., Berdak, O. (2016). *The downside of digital labs: digital financial innovation needs an end-to-end Agile and DevOps environment*. Cambridge, MA: Forrester.

Goldratt, E. M., Cox, J. (2004). *The goal: A process of ongoing improvement* (Third Rev. ed.). Great Barrington, MA: North River Press.

Gothelf, J. (2014). *Bring Agile to the whole organization*. Harvard Business Review. Retrieved June 24, 2016, from https://hbr.org/2014/11/bring-Agile-to-the-whole-organization.

Greene, B. (2004). *The fabric of the cosmos*. Random House Audio.

Griffin, R., Moorhead, G. (2012). *Organizational behavior* (10th ed.). Canada: South-Western, Cengage Learning.

Hastie, J., Wojewoda, S. (2015). *Standish group 2015 CHAOS report – Q&A with Jennifer Lynch*. InfoQ. Retrieved June 24, 2016, from https://www.infoq.com/articles/standish-chaos-2015.

Hayes, J. (2010). *The theory and practice of change management* (3rd ed.). New York, NY: Macmillan.

Huxley, A. (2006). *Brave new world*. New York: Perennial Classics.

Johnson, S., Cheng, R., Misiazek, S., Greytak, S., Boston, J. (2011). *The business case for agile methods*. Arlington, Virginia: Association for Enterprise Information.

Katzenback, J. Smith, D. (1993). *The wisdom of teams.* United States: Harper Business.

Krieger, J. *Creating a culture of innovation*. Retrieved October 8, 2012, from http://search.ebscohost.com/

Lapham, M. A., Williams, R., Hammons, C., Burton, D., Schenker, A. (2010). *Considerations for using agile in DoD acquisition*. Hanscom AFB, MA: Carnegie Mellon University – Software Engineering Institute.

Lathrop, D. Interviewed by Robert Weidner October 8, 2012.

Lewin, K. (1947). New frontiers in group dynamics, in D. Cartwright (ed.) (1952) *Field Theory in social science*. London: Social Science Paperbacks.

Lincoln, A. (1863). *The Gettysburg address*. Public speech.

McElroy, Morrow. *Employee reactions to office redesign*. Retrieved October 8, 2012, from http://search.ebscohost.com/.

Miller, G. *The magical number seven, plus or minus two: Some limits on our capacity for processing information*.

North Carolina State University, The Center for Universal Design. *Universal design principles*. Retrieved August 5, 2016, from https://www.ncsu.edu/ncsu/design/cud/.

PayPal. (2015). *PayPal enterprise transformation*. Retrieved July 27, 2016, from

Pierce, C. (1898). *The logic of events*. Retrieved August 5, 2016, from http://www.peirce.org/.

Pink, D. (2009). *Drive: The surprising truth about what motivates us*. New York: Penguin Group.

Puppet. *2016 state of DevOps report*. Retrieved July 27, 2016, from https://puppet.com/resources/white-paper/2016-state-of-devops-report.

Sayles, L. (1964). *Managerial behavior; administration in complex organizations.* New York: McGraw-Hill.

Scaled Agile Framework. *Case studies*. Retrieved June 24, 2016, from http://www.scaledAgileframework.com/case-studies/.

Schlitz, G. (2013, June 4). *Five facets of the agile organization: holistic change for the serious.* Retrieved March 2, 2014, from http://www.bigvisible.com/2013/06/five-facets-of-the-Agile-organization-holistic-change-for-the-serious/.

Schwaber, K., Leffingwell, D., Smits, H. (2005). *A CIO's playbook for adopting the Scrum method of achieving software agility.* United States: Rally Software Development Corp., & Scrum Alliance.

Scrum Alliance. *The 2015 state of Scrum report*. Retrieved June 24, 2016, from https://www.Scrumalliance.org/why-Scrum/state-of-Scrum-report/2015-state-of-Scrum.

Stenbeck, J. (2013). *PMI-ACP and certified Scrum professional exam prep and desk reference.* La Mesa, CA: GR8PM.

Sutherland, J. (2014). *Scrum: The art of doing twice the work in half the time*. New York, NY: Crown Business.

Sutherland, J., Schwaber, K. (2016). *The Scrum guide – the definitive guide to Scrum: the rules of the game*. Retrieved from http://www.Scrumguides.org/.

Sutherland, J., Schwaber, K., Thomas, D., Mellor, S., Martin, R. C., Marick, B., Kern, J., Jeffries, R., Hunt, A., Highsmith, J., Grenning, J., Fowler, M., Cunningham, W., Cockburn, A., Bennekum, A., Beedle, M., Beck., K. (2001). *Manifesto for agile software development*. Retrieved from http://www.Agilemanifesto.org/.

Takeuchi, H., Nonaka, I. (1986). *The new new product development game*. Harvard Business Review. Retrieved June 24, 2016 from https://hbr.org/1986/01/the-new-new-product-development-game.

Takeuchi, H., Rigby, D., Sutherland, J. (2016). *Embracing agile*. Harvard Business Review. Retrieved June 24, 2016 from https://hbr.org/2016/05/embracing-Agile.

Tillich, P. (1952). *The courage to be*. London: Yale University Press.

Toffler, A. (1970). *Future shock.* United States: Random House.

Thomke, S., Reinersten, D. (2012). *Six myths of product development*. Retrieved June 24, 2016, from https://hbr.org/2012/05/six-myths-of-product-development.

VersionOne. *10th annual state of Agile report*. Retrieved June 24, 2016, from http://stateofAgile.versionone.com/.

Wilber, K. (1998). *The essential*. Boston: Shamballa Press.

Yalom, I. (2000). *Love's executioner: And other tales of psychotherapy*. New York: Perennial Classics.