The Agile Code:

Principles of Leading Agile Teams

by JL Heather & Preston Chandler

with David Mitchell as editor

The Agile Code:

Principles of Leading Agile Teams

Preface

This is a book for business, marketing and IT leaders who have started, or are considering starting, an Agile transformation — leaders who need clarification, through real-life cases, on the benefits of Agile, along with the obstacles they may encounter through transformation. Leaders need to see how Agile can improve their work as well as prepare themselves to overcome common adoption struggles.

We have been working at VMLY&R, a digital marketing agency, for a number of years now, and we regularly find ourselves conversing with leaders in client organizations about the benefits, and obstacles, of an Agile transformation. Many of these leaders have already been working with management consultants to adopt Agile practices within their companies. All too often, these leaders are jaded about the promise of Agile, frustrated by dogmatic evangelists, and unsure if Agile is right for them. This perspective of leaders is unfortunate because we've witnessed the impact when Agile transformations are done right. We've seen, just like many experts,[1] teams double[2] their productivity, cut their time to market in half[3] and reduce quality defects by as much as 50%.[4]

So why does the jaded perspective exist? We think it is because transformations are tough. Not just Agile transformations, but all transformations. They fail far more often

than they succeed. Leaders may be sold on the idea of Agile and not understand how much work it takes to be successful. They have likely seen the tangible artifacts of Agile like sprints, kanban boards and stories that on the surface seem easy. They may assume that because they seem easy, getting a certification or copying what someone else did will be successful. It's important to find coaches or consultants who can go beyond the certification, packaged solutions, and cookie cutter methodologies so that you get a way of working specific to your company's culture.

We don't think that certifications and packaged solutions are necessarily bad. They fill a very important role in addressing business needs, but it is important to have an experienced coach who can help you navigate the waters of your own transformational rapids.

Effective transformations also take time and effort. Success is the product of years of effort from all across the organization. From every team. From every leader. It isn't something that happens over night, and it takes serious strength of will to keep going.

While the leaders we engage with are commonly in marketing and IT, we also speak with leaders from operations, strategy, design, and quality. They've often heard about the business benefits of an Agile mindset and doubt that the claims could be true. The claims of decreased time to market and improved productivity feel unbelievable, but we have personally seen teams reap even greater benefits. One marketing team changed their turnaround time for social posts from 25 days to 24 hours. The team that helped Preston learn about Agile in the first place had struggled for two years to launch a product and still hadn't released anything. After adopting Agile, they decided to start from scratch and released sellable, working software with the same feature set in six months. Another team found their quality defects fall by almost 90%. Almost every team that we've worked with has realized extraordinary benefits in some area.

Amazing results like these are definitely possible, but it's important to remember Agile is not a silver bullet. Adopting an Agile mindset is difficult and often at odds with existing cultures and ways of working. It's so different that many leaders don't see how their organization could make the necessary changes. We see leaders who are embroiled

in the midst of an Agile transformation, struggling to reconcile their traditional project and financial management practices with a new, more flexible, way of working.

After answering so many questions, participating in countless discussions, and working with so many partners, we figured it was probably time for a book. We wanted to share a number of principles that we feel are valuable in guiding the right behaviors. Since Agile can be a bit counterintuitive, we thought that using some real-life examples would help to illustrate the principles.

Our purpose is to assist you in asking the right questions and to help you look at your situation with a different perspective. While we do touch on some soft skills, we recommend that you consult books like "Crucial Conversations,"[5] "Good Authority"[6] and "Radical Candor."[7] Soft skills aren't our main focus.

This book is organized to ground you in a new paradigm before sharing examples. We wanted the reader to understand the basic principles of Agile, the relationship between those principles, the myriad of possible tools, and the system where those tools reside.

To begin with, these are the values that guide Agile principles and practices:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

— *Agile Manifesto, 2001*[8]

That is, while there is value in the items on the right, we value the items on the left more. The only adjustment that we commonly make to the manifesto is on the second value. We believe that Agile values and principles apply to every situation where a complex problem needs to be solved, so we reframe this value as "**Valuable outcomes**

over comprehensive outputs."

Reasons to Go Agile

The most common purposes, or benefits, of an Agile transition are that you want to build the right thing, want to build the thing right, or you want to build it fast, with heavy emphasis on building fast.

Principles, Systems and Tools

Principles, like those in the Agile Manifesto, serve to guide the creation of Agile tools. Those tools are used together to form systems for getting work done.

Understanding Your Journey

We use a matrix to describe where companies and teams are in their journey by evaluating their maturity along the dimensions of experimentation and operations. While significant benefit can be found by improving either dimension, the greatest value is unlocked by balancing both.

Problems Agile Solves

There are problems that you may already experience that Agile can help you with. Some of the most common problems are slow time to market, changing environments and poor productivity. We crafted three composite case studies to highlight each of these problems and how to address them with an Agile mindset.

Overcoming Obstacles

As with any change, transforming to Agile is fraught with obstacles. A few common obstacles are required documentation or process, management control, and partner adoption. We've again crafted three composite case studies.

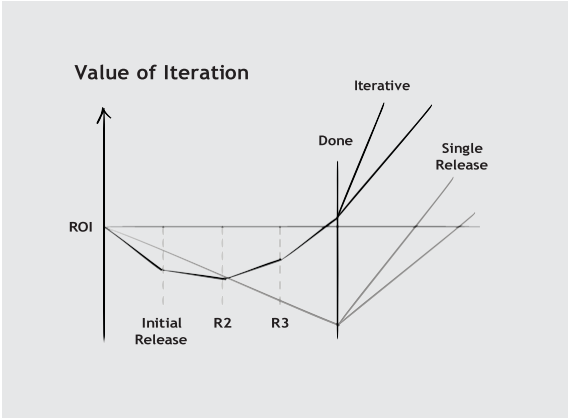
The Moral of the Story

If you don't have any time, then read this section. It is an executive overview and will provide several general principles to follow.

Reasons to Go Agile

As a business leader, you face complex decisions every day. You are constantly balancing costs and benefits, seeking the best combination. First, you may want to reduce the time that it takes to go from idea generation to idea implementation. Your industry may call this "time to market," or you may have some other term like "reducing lead time." Either way, getting something out the door faster is one of the major drivers of Lean and Agile transformations. Another area of focus may be ensuring the effectiveness of your solutions or products. That is, do they actually do their job well and do people like it? Lastly, you may need to increase the efficiency of your delivery so that it costs less to build, market or ship your solution.

If you are in any way involved in business, then chances are that you are currently facing one or all of these obstacles. You're in luck! Lean and Agile exist to address these issues in ways that are sustainable. Each of these pain points have corresponding Agile principles and practices that can be applied.



A faster time to market can be achieved by reducing the number of features in each release and limiting the number of releases that are concurrently in the system.

While doing this, we attempt to adhere to the following principles:

- Simplicity: The art of maximizing the amount of work not done is essential.
- Frequency: We deliver solutions frequently, often on short timelines.
- Agility: Continuous attention to results, excellence and smart design enhances agility.

This means that we carefully review the requested features and determine the smallest amount that constitutes additional value for the end user. Remember, 80% of your value will come from 20% of your features. For us, figuring out what not to include is just as important (if not more so) than determining what to include. While we do focus on having small releases, that doesn't mean that we can skimp on building a quality product. If we go down that route, we quickly find ourselves buried under technical debt and unable to deliver additional features.

We all think that we can predict the future, but the fact is that humans are rather poor forecasters. It took centuries of scientific research to allow a trained meteorologist to give a forecast that was more accurate than just looking at what the weather was like yesterday. We are best able to tell what is valuable to the customer when we give it to them. Unfortunately, this means that we have to expend effort without knowing for sure that the effort will prove fruitful. The end result is that continuously building iterative solutions provides the best value to customers with the lowest risk of failure.

Here are the applicable Agile principles that we attempt to follow:

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable solutions, ideas, and/or content.
- We harness change throughout the process to support our customers' competitive advantage.
- Complete, functional solutions, ideas and/or content are the primary measure of progress.

We don't know anybody in a finance department who wants projects to be more costly. In fact, we are being constantly pushed for greater efficiency and higher productivity, and that trend isn't likely to stop. We've found that cross-functional, self-organizing teams

provide the most efficient delivery of value. Many people assume that by dividing out the work, through careful orchestration, they can achieve high efficiency — but it is simply not true. Get the appropriate resource mix on a team and let them run with it. They will beat out the traditional project manager almost every time. At the same time, there needs to be a perspective of long-term sustainable pace to avoid team burnout. Replacing resources is really expensive.

Here are the principles:

- The best solutions emerge from self-organizing teams.
- We build projects around motivated individuals. Give them the tools and environment they need and trust them to get the job done.
- Agile processes promote sustainable work. The extended project team should be able to maintain a constant pace indefinitely.

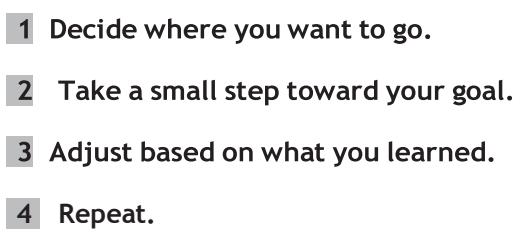
One of the values in the Agile Manifesto is, "Customer Collaboration over Contract Negotiation." That is, we believe that collaborating as human beings provides a unique understanding and solutions in a more valuable way than locking in agreements. Collaboration allows us the opportunity to continually improve. It sparks creativity and insight. It's part of being human.

There are a number of principles that target collaboration:

- We collaborate with our clients daily throughout the project.
- The most efficient and effective method of conveying information is through face-to-face conversation (in-person or virtual solutions).
- At regular intervals, our teams reflect on how to become more effective and adjust accordingly.

What is “Agile”? We came to a realization that agility is not a complex methodology or framework, but really just a simple set of rules.

Figure 1



What does it mean to be Agile, and why do we care when we have methodologies like Scrum, XP, and Kanban to light the way?

JL started as a Scrum believer and eventual master, certified and everything. Scrum was the answer to his questions and the pure definition of Agile. Slightly later in his career, he was exposed to a Kanban methodology that did away with some of the core tenets of Scrum and, after a bitter battle, his belief in Scrum had to bend because the Kanban methodology worked really well. There was more than one way to be Agile. If there were two ways to be Agile, what other ways existed? The short answer: a lot. JL read a lot of books, tried a lot of things (and made a lot of mistakes), watched YouTube videos, and started to have a really hard time defining what it meant to be Agile because there are so many ways to be Agile.

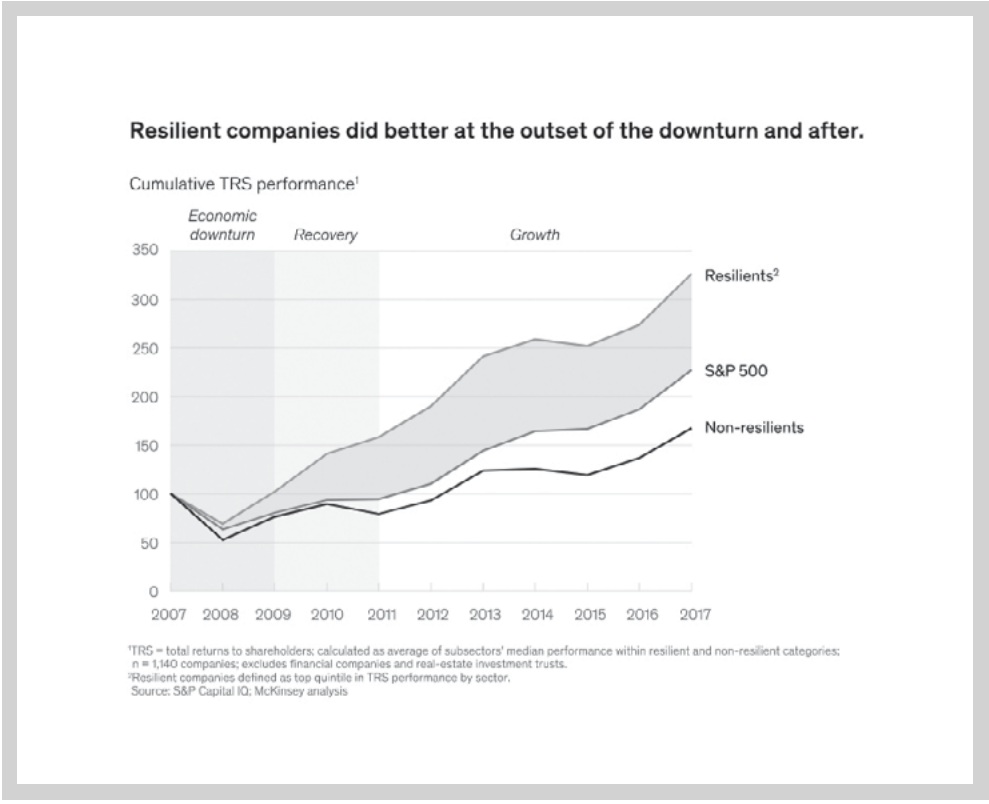
He eventually came to the understanding that if you follow the four steps listed above, then you are working in an Agile way.

That’s it. That’s all it takes to be Agile. No certifications, no consultants, and just a few simple steps.

Why Do We Care?

The fact that Agile and resilient companies thrive is foremost. If you aren’t Agile and

resilient, then eventually you go the way of the dinosaurs when a competitor that is Agile and resilient comes along. Agile provides a mindset and key principles that foster an environment of continual learning. That learning, in turn, drives an ever-improving system that maximizes value. Both value for the end customer as well as value for the business.



Just using an Agile methodology, however, isn't enough. Our most common coaching opportunities arise from organizations that have attempted to copy an existing Agile methodology like Scrum or Kanban. They assumed that by applying a specific process that someone else succeeded with, they too will find success. By the time we get involved, the organization is likely frustrated with their current approach and may even be saying that Agile doesn't work in their industry. Without meaning to, they have uncovered the truth that using a tool or practice without understanding it leads to a poor use of the tool. While we can certainly look at practices and find patterns of behavior that result

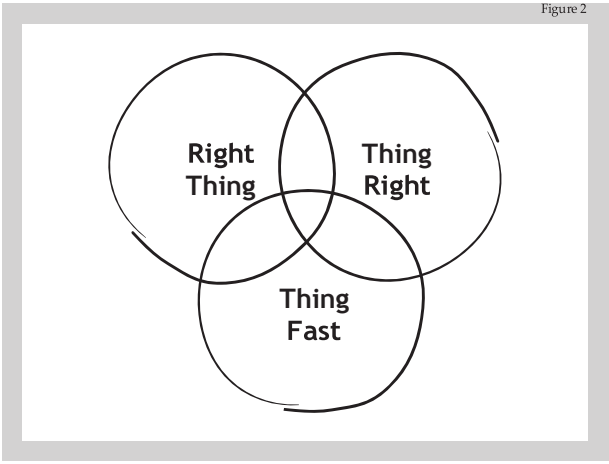
in better outcomes, each organization and individual is unique, and deserves a unique approach. The thing that truly drives the benefits of Agile is a learning mindset, not a specific process or tool.

Our focus in this book is not to teach tools and processes. Nor is it to get you certified. We want to expand your breadth of knowledge of Agile principles. Some of the case studies may strike close to home. They may provide key insights and aha moments. If that happens then we've succeeded.

Even though we are experts, we aren't necessarily experts in your company culture. Please take your insights and search for ways to apply the underlying principles instead of directly copying our solutions to the problems outlined in this book. It's our objective to guide you to a solution that works for you — something that you have ownership in and can maintain and grow, encoding it into your culture.

Problems Agile Solves

Figure 2



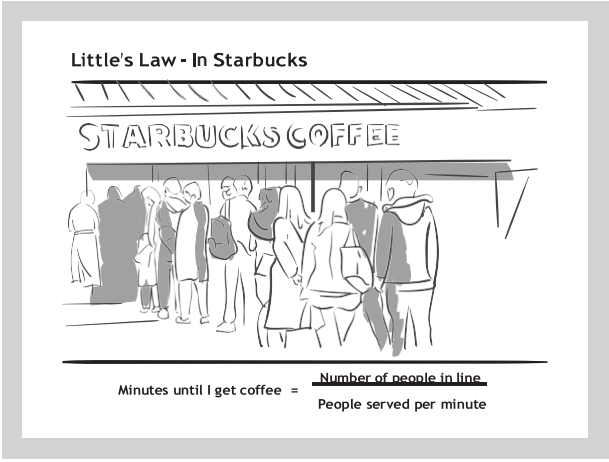
Pretty much every problem that Agile solves can be related back to the three main reasons to go Agile. While we will dive deeper into each of these problems in later chapters, we are introducing them to you here with a CliffsNotes version. This should enable you to quickly jump to the problem or problems that you are currently facing.

The first area is building things fast, or speeding up time to market. You may not call your problem "time to market though"; you may use other phrases like:

- The business or commercial environment is changing too fast, and I just can't keep up with it (attempting too large of a change).
- My IT department can't deliver anything quickly. It might take six months for the simplest of requests/changes (too many things in the system).
- Projects are always delayed, which means I can't plan because I can't predict when things will get done (controlling projects versus controlling systems).

We know, we know, they don't all look like the same problem, but they are all related to a shorter time from idea to launch.

In general, Agile makes use of Little's Law,[9] which indirectly states that the more, or larger, things are in a system, the longer it takes to get through the system. The way that we make the system faster is by breaking things into smaller pieces and then putting fewer of these pieces into the system at the same time.



The second area of problems that Agile addresses is building the right thing. However, Agile is focused on speed of delivery, so we are going to focus not only on building the right thing, but also on building it fast (iterative).

Some of the related problems that you might see are:

- The solutions that are delivered don't often have the expected outcomes (testable iteration).
- I need to prove ROI before I can get a budget approved for a project (long timelines).

The solution for both of these problems is iterative development. In conjunction with breaking things into smaller pieces, we need to iteratively validate and build upon what has been developed. This will allow us to try out the smallest item of potential value to see if it is actually "valuable," rapidly proving (or disproving) ROI and ensuring that the desired outcomes are achieved.

The last area is building the thing right. Again, we are going to focus on speed, so it is really building the thing right quickly.

- I have to cut costs every year (building efficiencies between functions).
- How do I find the right talent (retaining, empowering talent)?

While these two problems appear very different, they are both addressed by an empowered, cross-functional team — albeit for different reasons. Effective cross-functional teams naturally reduce the number of handoffs and friction between various functions, creating the most efficient use of resources to accomplish a goal. In addition, members of CFTs experience greater ownership, job satisfaction and lower turnover, making it easier to keep as well as attract the right talent.[10]

Principles, Systems and Tools

Principles, systems and tools describe how the world works, we promise! Once you start looking for them, you will see principles, systems and tools in everything. Even better, they give you a great framework for targeted improvement for almost anything! What, you might ask though, are principles, systems and tools? All right, we're sure you know the words, but here's what we mean in this context:

Figure 3

1. **Principles: Broadly applicable guidelines that help facilitate good decision-making.**
2. **Systems: Application of principles through a specific framework, process, etc.**
3. **Tools: The things you use to get work done within your system.**

That seems pretty simple, and it can be, but let's take a look at an example and use it to drive the point home.

Principles, Systems and Tools From a Doctor's Perspective

For the purposes of this example, we're going to assume you've been to a doctor a few times in your life (if we're wrong, don't tell us — just go get a checkup!). We'll further assume you've heard the dramatic line from the movies, "First, do no harm," represented as the Hippocratic Oath. While the Hippocratic Oath isn't quite that simple, in general it covers roughly four basic areas.

1. **Put the patient's best interests first.**
2. **Don't discriminate.**
3. **Prevent disease whenever possible, because prevention is better than curing.**
4. **Share your knowledge.**

Note: These are our words and a gross oversimplification of a very solemn and meaningful oath for the purposes of illustration. If you're a doctor ... sorry.

So, it's probably safe to assume that not only have you been to see "a" doctor, but you've also been to see multiple different types of doctors. The neat thing here is, they all follow some version of the above — at least if you visited a medical doctor. The oaths doctors swear are broadly applicable principles, and they help doctors make good decisions in the many tense, if not life and death, situations in which they may find themselves. It's easy! If you need to get going so you don't miss your movie, doesn't matter, see principle 1. If the person on the bed in front of you has a different color of skin, doesn't matter, see principle 2.

Even better, as a patient, knowing that your doctor swore an oath helps you trust them to treat you. You know that the doctor bending over you is there to make sure you are okay, which means you are more likely to cooperate, share information and trust. These principles make understanding and trusting a system easier, and that is great for doctors

and organizations.

Now, just because you know that your doctor swore the right oath doesn't mean you are going to trust a proctologist to treat your asthma. Those problems are on the opposite ends of the ... spectrum. What we're saying is that you are going to look for a doctor with the right skill set or knowledge to treat your specific condition. The same principles apply, but you are going to look for the right system for your situation. There are a lot of systems, so sometimes you might even need to see a generalist (for example, a coach or a mentor) to help you pick the right one.

Once you've found the right doctor, you may find that they use many of the same tools your other doctors find useful. Regardless of the type of doctor you see, any of them might send you to get tests done, have X-rays taken, or may poke and prod you with a fairly common set of tools. Outside of a few specializations, most doctors pull from a similar tool set even if they use those tools differently. So, while there's generally a lot of overlap in tools, take the example of a dentist (dentists are doctors too!), and you'll see very little overlap in tools. For example, if your dermatologist came at you with one of the picks a dentist uses, you might have a violent response. Picking the right tools to support a given system is critical to success.

Everything we talked about above, however, refers to physical tools you can touch or hold in your hand, but those are not the only types of tools we need to look at when

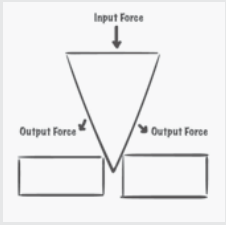
analyzing a system. We might also look at things such as sanitation procedures, like how a doctor always uses the little dispenser of hand sanitizer anytime they enter or leave a room, or the way they approach patient interviews or their take on a good bedside manner. For many different specialists who perform surgery, there are common preparation protocols that are followed to help ensure positive outcomes for the patient; these protocols are also tools in a system.

So, to summarize:

- Principles: The oaths doctors swear
- Systems: Dermatologist, pediatrician, ophthalmologist, etc.
- Tools: X-rays, MRIs, drills and picks (e.g., dental tools), scopes, diagnostic procedures, and more

Consider this more basic example:

- Principle: Wedges
- System: Hammer and nails
- Tools: Claw hammer (as opposed to a sledge or other hammer) and nails (as opposed to a screw)



Principles


Systems


Tools

From an Agile perspective it might look like this:

- Principles: Agile value statements and principles
- Systems: Scrum, Kanban, cross-functional teams, extreme programming, SAFe and more
- Tools: User stories, acceptance criteria, JIRA, Trello, journey maps and more

Hopefully this helps cement the concept of principles, systems, and tools, and you are now ready to take the next step. This is important because understanding helps you ask the right questions and push for the right action when your system isn't getting the results you need. This is true even if you aren't an expert in those systems. Let's take a look at where problems start to creep in.

Where Problems Start

Every team we've worked with, even the wildly successful ones, have problem areas where they can improve. These "problems" could more correctly be viewed as "opportunities" because you may not even see a negative impact when the teams are productive and efficient. However, you will see a positive impact as you better align your principles, systems and tools.

Problem 1: Not Understanding the Principles

When was the last time you asked everyone you work with why your company exists? If you did it now, how many different answers do you think you'd get?

If you aren't confident that the people working with you understand why the company exists, then it's probably also reasonable to assume that most aren't clear on the principles that should be the foundation for the way you work. Even if everyone is familiar with the principles, there is probably a variety of different interpretations based on what employees hear and see being practiced by their managers and coworkers day to day.

Let's look at the example of doctors from above. The first principle we have is to "Put the patient's best interests first," which most would take to mean that a doctor must put the interests of their patient first. Let's imagine for a second that we have a doctor researching the cure to some nasty disease, and they take it to mean they should "Put the patients' best interests first."

See the difference? It's subtle.

We went from "patient" to "patients," meaning this doctor is more likely to be doing what is best for all potential or future patients instead of the individual they are treating. What bad practices or questionable moral decisions might this allow?

Look at your coworkers and employees. How many of them will make decisions that are not aligned with your company's principles as you understand them? How sure are you that you are aligned with your boss's interpretation? What does this mean for your customers, and how could clarity improve your customers' experience with your brand and products?

The great news is that you can fix this lack of clarity and alignment. It's not easy, but it really just involves communication, open discussion and acting in line with your principles. Memos, posters, emails, and all that help with awareness, but it's the discussion and practice that truly drives understanding and adoption. That sounds easy, but trust us, it involves a level of communication and candor that isn't common in the business world. It's also important that, while it comes from the top, it must also come from every level of management if it's going to be effective in the long run.

Problem 2: Blindly Adopting a System

The most common problem we see as companies and teams adopt lean Agile approaches is that they find a preexisting system or tool that seems to work for someone else and try to implement it at home. Sometimes this takes the form of adopting a tool (i.e., JIRA, TFS, etc.). Other times it is looking at systems like SAFe[11] or the Spotify[12] model and thinking those systems will magically work in any situation. In either case,

you are letting someone who you have never met, and who has no understanding or exposure to your company and situation, drive how you operate. If that doesn't scare you, it should.

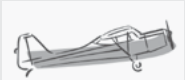
The perfect example here is cargo cults[13] — yup cults. Go on, google it; it's a thing. For our purposes, we use the term "cargo cult" to describe any situation where a person's understanding of cause and effect are reversed from reality.

Here are a couple real-life examples:

- Creating a runway and waving flags with the expectation of creating or summoning an airplane
- Crafting eyeglass frames out of sticks and expecting better vision or creating a wicker airplane and expecting to be able to fly.

Real

Mimicked



Real

Mimicked

- Using Agile terminology for traditional ways of working and expecting improved results

Now that you're done googling, let's go back and talk about dentists. Let's imagine you are leaning back in the chair and you ask your dentist where they went to school. The Now that you're done googling, let's go back and talk about dentists. Let's imagine you are leaning back in the chair and you ask your dentist where they went to school. The dentist calmly picks up the drill and, as they put the tip against a tooth they say,"Oh, I didn't go to school, but I've watched the dentist over there do this a lot." In this situation, we'd like to time how long it takes you to hit the door on your way out.

How does this apply to us? Well, this wannabe dentist is assuming that all there is to being a dentist is blindly doing what other dentists do. Don't think this happens in the

business world? How many times have you heard something like this: "I just heard that ACME is doing Crazy Cool Process and they are leading their industry; we should try that here." There is no understanding of why Crazy Cool worked for ACME. There's been no analysis of how the situations at ACME are, or are not, similar to yours. There's probably been very little thought at all, just reaction.

This happens with Scrum, Kanban and other Agile methodologies all the time.

If people truly understand the relevant principles for their work, this kind of mistake is a lot harder to make. When there is no understanding of the principles, the "why," then this is a simple mistake to make in an honest attempt to make things better.

What can you do? Well, first, make sure your company or department principles are well understood by the ones looking to drive change (look at the previous problem). Again, when we say "well understood," we don't mean printed out on nice paper and posted around the office or tacked onto email signatures. We mean deeply discussed and lived on a daily basis. Next, use those principles to question any proposed changes or to evaluate any systems being considered.

Your best option is to find someone with broad lean Agile experience to help you build a system that works for your situation and is aligned with your principles and goals. While we would love for you to hire us, that's not the point here. Principles usually run into company politics. If you are attempting to change or reaffirm principles, you will likely need someone who doesn't sit in the middle of the organizational structure to champion them. Many executives need just as much, or more, coaching as frontline workers, so another executive to coach on principles often doesn't work. It normally requires someone who sits outside of the existing company politics. Yes, it can be helpful to hire a consultant.

Problem 3: Not Understanding the Tool or Using It Blindly

There are a lot of tools out there, and it's hard to know what tools work best in which situations. It's also hard to find the time to fully understand the tools available and make an informed choice about which tool fits your situation best. That's why it's so common

to see people and teams using tools blindly without a full understanding of the why. Best case, the teams aren't getting the full value of the tool being used. Worst case, the team blindly uses the tool and does more harm than good, causing confusion and limiting or preventing results.

1. **Just a bad tool**
2. **Wrong tool for the situation**
3. **Right tool used incorrectly**

The issues with tools come in three flavors:

Some tools just shouldn't be used. To fully abuse the doctor analogy, there's a reason leeches aren't considered a great form of treatment any longer. Hitting closer to home, some would say traditional waterfall-type approaches to problem-solving are outdated in the same way. We wouldn't go that far. Waterfall has its place, but in looking at any practice you will find tools that have been discarded because they have been found lacking when compared to modern practices.

Other tools are good and productive, but just being used in the wrong situation. For example, using a hammer to drive in screws will lead to more frustration than results. If you hear something along the lines of, "But, this is the way we've always done it," it's very possibly a good tool in the wrong situation.

Lastly, teams adopting change and trying to improve are very likely to pick up a solid tool and use it incorrectly. This isn't incompetence or stupidity; it is usually the exact opposite, but it is a good opportunity to provide guidance. Guidance is very often found in the form of a consultant versed in the application of the systems and tools your team is

Summing it up:

1. **Principles: Broadly applicable guidelines that help facilitate good decision-making**
2. **Systems: Application of principles through a specific framework, process, etc.**
3. **Tools: The things you use to get work done within your system**

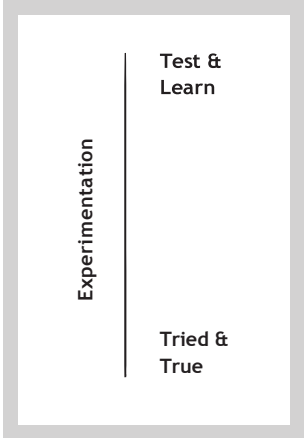
Common problems or opportunity areas:

1. **Not understanding principles**
2. **Blindly adopting a system**
3. **Not understanding the tool or using it blindly**

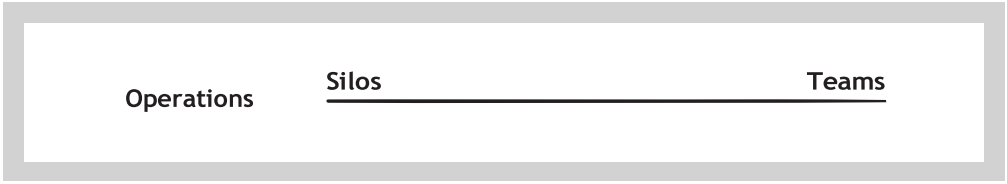
Understanding Your Journey

Everything is broken.

If you read that and thought it felt a little pessimistic, it might be time to shift perspective a little. If everything is broken, then anything can be fixed. Perfection is a goal that is pursued but never achieved, and the journey is about fixing the little or big things along the way. That said, the first step of any journey is understanding where you are now as well as where you want to be; it lets you know in which direction to take your first step. We've found it very helpful to start by evaluating the companies we've worked with in terms of how comfortable they are with experimentation and how well the organization executes or operates.



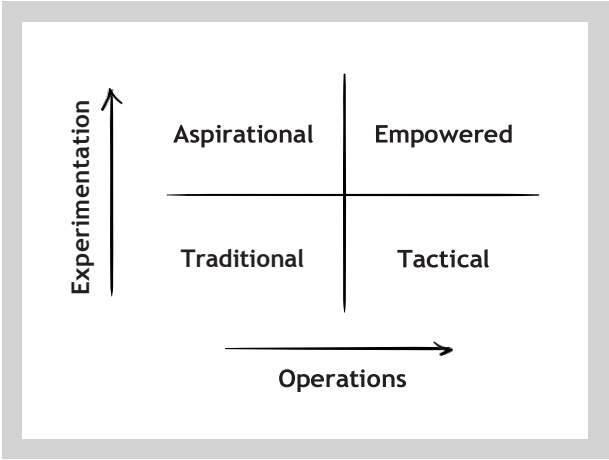
Experimentation is all about how open you are to trying new things or how willing you are to risk failure to learn. This is primarily in terms of your approach to product development but often also applies to how you evolve your process. For instance, your teams may be willing to test many new and creative solutions to your customers' problems while being very strict and risk adverse in their approach to delivery.



Operations, on the other hand, focuses primarily on the execution of an idea or concept rather than its initial conception. We see companies that still operate based on the tried-and-true waterfall approach along with companies that have found an individualized Agile approach that truly works for them. One of the strongest indicators of operations maturity is how much the environment leans toward teamwork rather than individual contribution. Maturity here takes on qualities like agility, speed and increased quality.

At f irst it might seem that there is little overlap between experimentation and operations, they seem like separate functions within an organization. You can create value by improving along either the experimentation or operations axis. What we hope you'll see by the end of this chapter, though, is that organizations who have achieved true success have done so by making experimentation a key factor in their operations. It's the combination of the two that creates true competitive and often disruptive value.

The concepts of experimentation and operations are tightly coupled and form a 2x2 matrix that we've found very useful in defining our customers' starting point in terms of four different archetypes: traditional, aspirational, tactical, and empowered.



Traditional organizations are characterized by the traditional ways that they operate and look at problems. There is nothing inherently wrong with traditional organizations, and in some industries a more traditional approach may work or even be required in certain aspects of doing business. That said, we typically hear clients state that a traditional approach is required in their case because ... well, insert a variety of reasons: legal considerations, safety, supply chain constraints, technological constraints, etc. We've heard it all, and we can tell you that in every situation we've experienced, there is room for more experimentation and agility.

More often than not, what you're fighting in traditional organizations is momentum. Complacency creates "this is the way we've always done it" mentalities. Silos exist because it is easier to create them than to break them down. Success in the past has created risk aversion in the now. People have built careers on the current SOP. And change is just scary; that's not joking — it's fact. People struggle to accept and embrace change. It usually takes a shock of some kind to motivate traditional organizations to change, and when they do they typically either look to innovate (i.e., experiment) or to become more efficient (i.e., Agile or lean).

Aspirational organizations are those that have gone down the path of experimentation, either at the expense of, or independent of, operations. Typically there is an investment in human-centered design (HCD) or design thinking practices. We are starting to form a better understanding of the customer that allows new innovations. We see some truly

great ideas come out of aspirational organizations, but those ideas rarely seem to get fully off the ground, and if they do, it's often too late.

Some aspirational organizations are blissfully unaware that their poor operations are keeping them from success. Others understand the need to improve how they work; they just don't know how to go about it. Some of our favorite engagements have been with these organizations that are extremely eager to change and are willing to do the hard work and make the tough decisions that a successful transformation requires.

Tactical organizations take the other path; they focus on their delivery or operational approach. Often we will see an organization looking to adopt a very specific Agile or lean methodology that has worked for a competitor or adjacent company. Ideally they are doing so with the help of coaches or other experts, and many times they will see some or even great success. That said, it's the organizations that evolve beyond the cookie-cutter methodologies and evolve an individualized approach to build something that is not only maintainable, but scalable.

These tactical organizations often operate very efficiently. Their projects are on time. Their budgets look good. Their people are productive. The problem is that the phrase "garbage in, garbage out" still applies. Without innovation and experimentation, they are delivering the same type of work they did before. The value of their solutions are not improving. That said, they are seeing success because they are getting to market faster with more predictable results, so the lack of innovation may not be immediately visible, at least not until they run up against a competitor that has mastered both experimentation and operations.

Empowered organizations are the very rare organizations that have managed to effectively weave experimentation throughout their operations. This means they are quickly and efficiently identifying customer value, prototyping and testing solutions, and delivering those solutions to customers while adjusting and tweaking for maximum value the whole time. The only constant is change. We know that you just read all that and thought it either sounds too good to be true or too far out there to be possible. It's neither.

Empowered organizations have some common characteristics, and, while they will differ wildly in the details, there is value in the principles. These are the foundational principles of this book, and the principles we hope will become part of your professional DNA by the time you're done reading. These are the principles that we are passionately pursuing with every client.

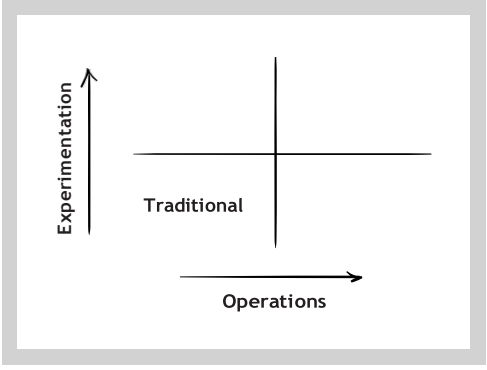
If you looked inside an empowered organization, the first things you might see are small, cross-functional, autonomous teams. You'd see teams owning problems from identification to solution delivery, commonly talking to real customers and testing out very rough solutions with small subsets of customers. You'd see work visualized everywhere. You'd see the process changing constantly. It would feel alien and wrong at first, scary even, but that is because it's just so different from how most of us work — but it's great, freeing, and insanely valuable.

At this point, it's important to call out that there are no clear lines between these four archetypes. An organization can be tactical in some ways, traditional in others, and aspirational in other situations. Sometimes different parts or divisions may have vastly different approaches. All of that is to be expected. These archetypes are a tool to help guide conversations and define vision.

The case studies and examples in subsequent chapters will highlight how these different archetypes can improve and change. If you feel like the case studies are very familiar to you, and that they may refer to a specific organization, don't worry. The names and situations have all been adjusted to protect the innocent. These cases are composites and represent some of the most common situations that we've seen as we have coached teams. So yes, we hope they seem familiar because we see the same stories play out in countless companies, across all industries and job functions.

Problems Agile Solves:

Always Delayed, Never Fast Enough



Large Corp had been around for decades as a behemoth in the marketplace. Their management practices and key performance indicators (KPIs) were well entrenched and seldom changed. One of these key practices, annual planning, began a full year in advance of the year to be planned.

The annual plan called for projects that were always big and always complex. There was ample evidence that when they attempted large projects, they almost always failed in one way or another (late or never delivered, over budget, ineffective, etc.). Despite the facts, large projects, big releases, and complex solutions continued to be the norm. Why, when facing the facts, was it so hard to build the case for change?

Large Corp had complex legacy systems that had been built over decades and were prone to breaking. It took a lot of time and work to avoid customer-facing problems, and even then problems were common. When problems did make it in front of customers it resulted in a lot of extra time, effort and money. Given all this pain, it was only reasonable to try to avoid it and give people more time to ensure there were no issues. They did it in the easiest way they could: doing the things that caused the most pain less frequently, and to some extent that helped avoid the problems.

As an extension, Large Corp was organized in departmental silos, which added to the complexity around planning and execution. It wasn't just a single department or team that was planning — it was many. They had big complex systems, and they felt they needed specialized teams to handle the different layers of that system to make sure there were no bigger issues that resulted in what we lovingly call "CNN moments." You know, those moments when the CEO has to be interviewed on CNN to explain why something, like a data breach, occurred? Given Large Corp's complexity, generations of leadership had planned around, and executed on, big chunks of work to maximize the time the teams spent working. This helped avoid the pain of planning and made the difficult handoffs between departments fewer and farther between.

It always seemed reasonable for Large Corp to reduce "overhead" to reduce the complexity of the process, which unfortunately led to even bigger chunks of work and more complexity over time. With complex, fragmented and siloed leadership in addition to infrequent large handoffs, those doing the work rarely understood the reasons for the work. Instead of seeking understanding, they put their heads down and accomplished all of the tasks in front of them. When work did eventually get completed, it was handed off to the next silo with few people questioning if the work actually solved the problem at hand.

The large and complex projects were killing the business, burning their people out, and driving their customers away. It was a vicious cycle that needed interruption. Leaders, seeing no other choice, bit the bullet, brought in some new leadership and mandated a more Agile approach on an impossible timeline. It was good intent poorly realized, as

they lost sight of some key principles along the way.

Key principles that can apply in this situation are:

- Doing Less Gets More Done: Smaller batches and less work in progress (WIP) creates better visibility, more predictably, and produces work at a higher quality.
- People Want To Do Awesome Work: When people aren't doing awesome work or you feel like they don't want to do awesome work, look at the system first.
- Avoiding a Problem Is Not Fixing the Problem: Doing something less often may feel better at first but tends to make the problem worse overall.
- A Better Engine Doesn't Help if Your Tires Are Shot: The performance of the system as a whole is more important than the performance of any individual part.

Doing Less Gets More Done

Large Corp had good and clear KPIs, but that by itself wasn't enough. There was a prioritized backlog, but since there wasn't a clear indication of what would really move the needle on their KPIs, there was a big push to start everything and to get everything done. It was a shotgun approach, to do it all and hope something sticks. Hope isn't a plan, and what really happened is everything went slowly. It was like trying to cram thousands of cars on the highway all at once, and all we had was gridlock and frustration.

Rather than assuming everything had to be done and live for customers, we should have identified the two or three most valuable features, completed them rapidly, and used what we learned to pick the next most valuable features. Even if it turned out that those features weren't the exact right ones to start with, they would have been done and customers would have had some relief rather than none. Additionally, we would have ironed out many of the problems in the system along the way, so the next features would come even faster.

When you see a situation where work is piling up and timelines are extending, there are a few good questions to start asking your teams to help get them thinking in the right way.

- Are we finishing work at the same rate we are starting it? If not, why?
- What is preventing work from flowing smoothly through the system? Are there blockers, pain points or missing skill sets that need to be addressed?

People Want To Do Awesome Work

There were some really good people working with us at Large Corp, and they were very eager to find a better and less painful way to work. They had real hope that things were changing and they would be able to do their best work rather than fighting the system. That made the situation far more painful. We've seen that the larger the organization, the more risk-averse they become. The more risk-averse an organization is, the more time they spend on planning and approvals — and the longer it takes them to get anything done. Incidentally, the more approvals, the less innovation. You might start to get the feeling that we don't like formal approval processes, and you'd be right.

Large Corp had set up metrics to measure progress and performance. These metrics were not great or actionable metrics in many cases and were often used to blame and shame teams for performance problems they had no control over. As you can imagine, some of our best people left rather than face the abuse. Metrics were a driving force here, but not the only driving force.

Executives feared failure and did not trust teams to feel the same urgency to succeed. Their use of the metrics demonstrated that lack of trust to everyone on the project. Instead, leaders should have spent their time making their vision clear and concise, helping people understand the urgency, and practicing true servant leadership. Under the paradigm of servant leadership, metrics should have been used to find bottlenecks, remove blockers, ask questions, and to generally support (rather than blame) the teams.

Teams should have been held accountable to real business results rather than targets arbitrarily set by leaders who, while they understood the business problem, did not understand the process problems.

When you see morale suffering and people losing sight of the fact that they work with other people and not metrics, there are a few good questions to ask.

- How are the metrics we have helping us ask the right questions and better serve the team?
- Are the metrics we look at good indicators of progress and business value?
- What is preventing the teams from delivering direct business value?
- Are there any required skill sets not represented on the teams that own the work?
- Are we providing opportunities for our people to develop their current or new skill sets?

Avoiding a Problem Is Not Fixing the Problem

Large Corp had some good examples of fully facing a problem and fixing it. For instance, when we started out, releasing work to the public was a very painful process. There were a lot of approvals required (even though only a few were needed), and the technical process was complicated and risky. This was tackled head-on with the direction that we would be releasing code often, daily in some cases. As you can imagine, there was some heartburn over this initially. It was a painful process, but the team in charge of releases took ownership and started releasing often, and things, at least from a release perspective, started to smooth out. Before long releases were a nonevent, there wasn't stress about the release, even if there was some stress about the content. In the end, it helped other teams focus on what mattered — fixing problems — because they trusted there wouldn't be issues with the release.

It sounds fairly rosy as described above, but that is mostly because of the team in charge

of releases. They took ownership, and their lead had connections that let them tackle the problems they faced on their own. Their lead also fully believed in, and acted on, many of the principles outlined above.

Without this lead, and the support he brought to bear, things would have fallen apart. The lack of support from the leadership of the initiative meant this team was on their own and, while they were successful, some of their success was at the expense of productivity from other teams. For example, this team set their own goals, which were not always ideal for the overall delivery of value, and a leader at the initiative-level should have been there to ensure the system as a whole was optimized, rather than just the process of releasing to customers. This requires clear and relevant goals that ladder up to the goal of the overall initiative. Additionally, leaders should have worked to stop the separation of this team and promote full integration with cross-functional teams to ensure issues were being raised and resolved quickly and efficiently.

In many ways, the success of this team hid or shifted the problems they were tasked with solving. It's important for leaders to ask questions to ensure they are looking at root causes, rather than symptoms, to ensure problems have been solved and not just hidden or shifted upstream or downstream.

- Why? Why? Why? Why? Why? (if you aren't familiar with the concept of 5 whys,[14] take some time to look it up!)
- Help me understand the full impact of this issue so I can ensure it is getting all the visibility and support it needs. (we know, it's not really a question.)
- If something is painful and slow, is it necessary? If it is, how do we make it less painful and do it more often with smaller batches?
- What support does the team need to resolve this problem, and how can I help attain this support?
- As in the previous section, are there any needed skill sets not represented on the teams that own the work? If so, how do we build that skill set into the team or better coordinate with teams that already have that skill set?

A Better Engine Doesn't Help if Your Tires Are Shot

It was clear when we got to Large Corp that one of its departments was very used to getting its way. The work it did tended to have farther-reaching impacts, and the team had some strong connections at the top. We had initially pulled the members of the department into cross-functional teams, but at the first sign of stress they were pulled back to be "protected" and to ensure the quality and speed of their work. They were allowed to work as a separate departmental team at the expense of the other teams they left. The result was that a new and expensive set of handoffs was created, extensive documentation for work was required, and the team, feeling special, felt it could blame almost all its failings on the inputs of other teams. The departmental team was now optimized to do its work only when it had exactly what it said was needed, but this local optimization was far overshadowed by the burden it put on the rest of the system.

Additionally, shortly after this department pulled back from its cross-functional teams, other departments attempted to do the same with varying levels of success. The department had set an example, and a lack of response from leadership had implicitly endorsed the action.

We're sure, if you're like most people, you have a good idea of what we are going to say, and we're equally sure that you know there are "realities" of the situation that make all of this hard. That said, being a leader is hard, and these are the realities you have to fight.

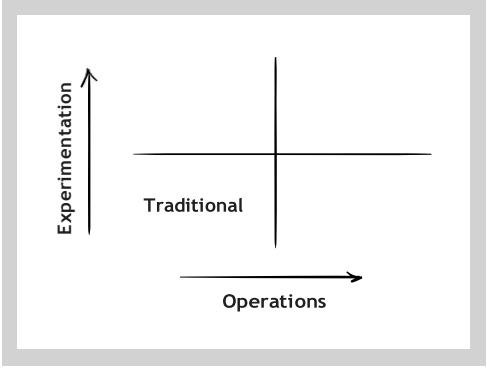
- Keep cross-functional and co-located teams whenever possible. If you have to pull people back or share resources, do everything you can to have them sit with the teams they support as often as possible. In the Large Corp example, the department team sat in a different room and then a different building. It would have been better if they sat with the teams they supported at least part time.
- Provide coaching, mentoring and other support for team members. Change is hard and scary and, while people want to do awesome work, they need support.
- Measure and show the impact. System-level metrics should have been used to escalate the problem and communicate the impact on our ability to deliver.

None of this is easy, and sometimes the best you can do is ask the right questions.

- Are teams optimizing their work at the expense of others? How are we measuring this impact?
- What handoffs do we have between teams that are slowing us down?

Problems Agile Solves:

Changing Environments



At Widgets Co, we were brought in to assist with the digital transformation of their customer-facing app and website properties. Knowing that its industry was in a constant state of flux, Widgets was very concerned with making sure that it was prepared for that future. In a balance to that desire, they had built products for decades that, if built or used incorrectly, could endanger people. This made them very averse to taking risks. As a result, Widgets enacted processes and policies that would reduce the possibility of failure, ensure no details were missed, and clearly communicate who was responsible for each aspect of each component.

Mark, a recently hired marketing manager at Widgets, self-identified as a technophile and was always looking for the next and best trends and opportunities to improve the

customer experience. His previous employer was a small tech startup, so Mark was pretty frustrated when he took his latest and greatest idea to his boss to get it approved, only to find that he would somehow have to prove the ROI in order to get a project approved. All he wanted to do was keep up with the ever-changing tech landscape. He didn't have the time to waste putting together justifications, especially when proving the ROI would be a project in and of itself.

Biting the bullet, Mark dutifully began to complete the project request form. To calculate the ROI, Mark needed to understand not only the potential return, but also the expected investment. To figure out the investment, Mark took a stab at designing the solution by himself and then, later, worked with other departments to determine the level of effort. After many meetings, including meetings to plan meetings, he finally completed the request form. Mark was really frustrated by how long it took, but his boss informed him that four months to finalize the request wasn't horrible — in fact, it was actually on the quick side. Now it just had to go into the queue to be prioritized and worked on.

Here are a number of key principles to follow:

- Planning Doesn't Predict the Future: More planning doesn't improve your ability to predict the future — we know, it doesn't sound right. "Everyone" tells us to plan. If you fail to plan, you plan to fail. I'm not telling you not to plan, but additional planning doesn't predict the future. It has taken countless datasets, scientific models, and fields of study just to predict the weather, and it still isn't all that accurate. It's no wonder why yesterday's weather[15] has been the best predictor of today's weather for thousands of years.
- Flexibility and Agility Are More Effective Than Planning: Since you can't predict the future with a good degree of accuracy, the best alternative is to be flexible to the changes that will inevitably come. Be like the cockroach, able to survive most natural disasters, survive without eating for weeks, and regenerate limbs.
- Trying Out Real Stuff Reduces Risk Faster/Better Than Planning: Hoping that something works versus seeing something work firsthand makes a dramatic difference in our learning. Strike that — building a real thing and trying it out is learning. It's real learning from reality. You can't beat that with planning.
- Fixed Scope Is a Myth: A "fixed scope" rarely stays fixed and even more rarely results in a fixed timeline/budget.

Planning Doesn't Predict the Future

Do you require proof of ROI, or even projected ROI, to get a project approved? Unless you've already done it, then you can't have proof. You can, however, have evidence. Evidence by nature is not conclusive, but it may provide enough of an insight to make an educated guess. However, it is still a guess. Anecdotal evidence like external research, hearsay, and conjecture is quite weak in its ability to form conclusions like planning and predicting the future. While you can, and should, require evidence of opportunities and potential success, you should recognize them for what they aren't — proof.

How much planning is required to get a project approved? We've seen many organizations that require detailed plans and designs prior to getting a project approved. That is half the project before the project even starts. And what is the team planning

around? Project plans are fairly static documents. Does the team have enough evidence to support the level of detail that their plan implies?

Flexibility and Agility Are More Effective Than Planning

Are changes in scope expected? Or punished? Organizations that don't expect scope changes, or strongly discourage changes, are placing more value on the plan than they are on a successful outcome. They believe, erroneously, that controlling the plan will ensure the outcome. That may indeed be true in a world where nothing else changes. In reality, we know that the world is constantly changing. Rather than locking in a particular decision, we will be much better served by finding the ways to remain flexible in our planning and solutions for as long as possible. This ensures that we are prepared to adjust to external changes or learnings as they occur rather than waiting for the next project.

How much value do you place on flexibility? Do you prioritize decisions, systems and tools that will allow for additional flexibility? When confronted with multiple options, do you choose the option with the most f lexibility, even if it has a bit less immediate value? You can't predict the future, so you might as well prepare yourself for the inevitable changes that will be required to adapt to the future.

Trying Out Real Stuff Reduces Risk Faster/Better Than Planning

Did you assume that risk was eliminated because you planned? Many companies, like Widgets, assume that additional planning can eliminate the risk on a project. We love the quote from Helmuth von Moltke the Elder, "No plan survives contact with the enemy." Or, this one from Mike Tyson: "Everyone has a plan until they get punched in the face." While there is value in planning, and usefulness in understanding risks, planning doesn't change the nature of the risk itself — only actions do.

How are you validating planning assumptions? One method to validate assumptions and reduce risk is to conduct a Design Sprint. If Mark and the team had done one, it might have looked like this:

Biting the bullet, Mark gathered a small, cross-functional team and embarked on a one-week workshop to clarify the core problem, select the best options, and test them. Together they dove deep on understanding what the customer needed and why. In the first half of the week they didn't even create any solutions — just more problems. They pulled internal data, as well as external research, to help them understand how big the problem was, and how much opportunity there was for improvement/profitability.

Toward the end of the week, the team crafted a simple solution that would allow them to test out many of their assumptions. Assumptions about the problem, about the customer, and about the solution all needed to be verified. What they found astonished them. Many of their assumptions were disproven, and new insights were gathered. Mark felt that, while he understood the problem better, he still didn't understand enough to move forward with a full project. He shared what the team had uncovered with his boss, and they discussed the value of an additional design sprint.

How are you reducing risk? In general, the longer risks exist, the larger the impact. Our goal should be to identify and reduce risk as early as possible in the project. In fact, you may prioritize risky items over valuable items early in the project because it makes the long-term development much more consistent. How do you prove that a risk is reduced? If the answer is anything other than testing it out, you've likely not reduced the risk.

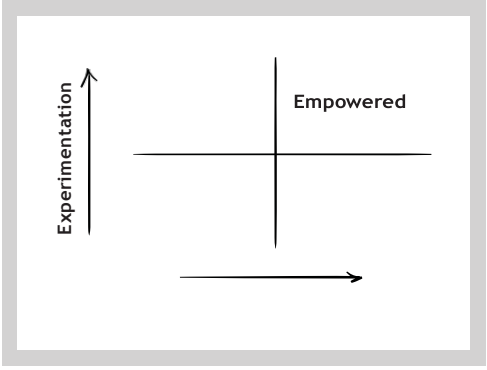
Fixed Scope Is a Myth

Do you fix scopes? Chances are that you do. You've likely been convinced throughout your entire career that scopes should be fixed. It's just how things work, isn't it? The best example here is the countless essays that you were assigned to complete throughout school. How many of them required a certain number of pages? Words? That is the fixed scope. You must use that many words (or release that many features) to be complete. What if, like me, you valued concise communication, instead of like JL, who values verbose communication? Well, logically you begin to play with the font size and margins to ensure that you hit the right number of pages without adding additional content.

How difficult is it to adjust scope? Does it take an act of Congress? Once the team has worked through the reasoning, it shouldn't take much more than changing a line on a spreadsheet or moving a ticket in your backlog. If it takes more effort than that, chances are you're not as prepared to change your scope as you should be.

Problems Agile Solves:

Fast and Cheap FTW (For the Win)?



We had been working with Global Tek for a long time, and things had fallen into a rhythm where the team thought it could start transitioning work to some lower-cost remote resources to make things faster and cheaper for our client while increasing our margin. Seemed pretty simple as the work wasn't really complex, but, as they so often don't in these situations, things just didn't work out.

They started transitioning the work and felt pretty smug since they had escaped the time zone issue by looking south instead of east or west. They even had people working in their remote office who spoke English well. It seemed like they had dodged all the bullets that so often doomed to failure remote work like this. They still had people sitting with the client every day; they had just switched things out on the back end and were confident

the client wouldn't even notice since they never talked to those guys anyway. It really did seem to have every possibility of working, so it was even more painful when it didn't.

At first, it seemed like things were going well. They were transitioning work and local resources were rolling off at a regular pace and no major issues were popping up. Soon, however, as they fell into a new rhythm, the frequent communication, reviews, and mentoring started to taper off, and it wasn't long before they had a couple small schedule issues. Not long after that, they had a launch that, while it wasn't late and the client was happy-ish, the quality was such that they had many issues to fix — hopefully, while the client was unaware there were any issues. As you can imagine, all of this quickly started to pile up.

Schedule and quality issues started to explode weekly or daily, and the client was understandably upset and possibly more so because they didn't understand why things had suddenly gone sideways. The work was quickly brought back to the home office like it had never left, and leadership and the team started apologizing and fixing in a never-ending cycle. Soon, we were burning through our budget at twice the sustainable rate with no hope of ever seeing that money come back. The client had learned that if they yelled enough we would apologize and give them work for free and as fast as possible.

It took months, more schedule issues, and even a new high in quality issues over the holidays before the team really started to turn things around. Overall, the client spent a lot more than they wanted to, and we wrote off a lot more money than we will most likely ever make back. All of that doesn't even account for the impact to customers and our client's business.

Here are a couple of key principles for this situation:

- Communication Isn't Always Scheduled: In terms of getting work done well and on time, unofficial and unscheduled communication is what makes it work — however it happens.
- Caring Decreases With Distance: While this is true geographically, we mean that the further you are from the problem or the less accountable you are for success, the less you care.
- The Best Functional Is Cross-functional: By "best" we mean the fastest and cheapest. Cross-functional, and ideally co-located, teams are almost always cheaper and faster overall.
- Only Be Sorry Once: Own your failure and move on to fixing the problem. Results are the best apology.

Communication Isn't Always Scheduled

We have a tendency to undervalue casual conversation, water cooler talk and the overheard tidbit because it all flies under our radar as part of daily life. A lot of it isn't even all that relevant to work. So, it's natural to think distance between team members won't make a huge difference. The truth of the matter is, and this was the case for Global Tek, that unscheduled communication is where much of the knowledge transfer happens. It's where teams become teams rather than just a collection of individuals. It's how we hold each other accountable and provide and accept feedback. It's how everyone stays clear about the goal and on the same page.

The key point in the story is where the new rhythm formed and the communication tapered off. It's hard, but when you have remote teams — or team members — you have to over-communicate. This isn't news, but it still happens almost every time with remote teams or team members.

Some common tactics that work, include:

- Everyone dials into calls: It cuts down on side conversations and puts everyone on equal footing
- Always-on video chat: It helps people feel more in the room even if it feels strange at first.
- Travel: It can be expensive, but bring the team together once a quarter or a couple times a year

If you're seeing issues with a remote team, or if you're getting ready to spin up a remote team, the actions above will go a long way toward preventing or fixing issues. There are also a few key questions to ask your team to help push them in the right direction.

- How well do you know the people in the remote location? Could you name all of them?
- What are you doing to help remote team members be part of the team, gain clarity on goals, and feel the same sense of urgency we feel here?
- Do you have a clear and documented process for handing off work and resolving issues?

Caring Decreases With Distance

The remote team working on Global Tek just didn't feel as accountable for results as the team sitting next to the client. There's no judgment in that statement — how could they? All they saw was a backlog of work and all they heard was someone saying it had to be done faster. There was nothing in the message about why. Even if they were just a hundred yards away rather than in another country, the results would have been the same. Because they didn't care as much quality suffered, they were less likely to push to hit a date, and lastly (most importantly), it created a rift between them and the local team.

This is one of those things that's easier to say than do, but you have to find a way to

share the why. Global Tek shared the work, but never talked about why it was important. The remote team never even heard our clients' voices. It wouldn't have been hard to set up time for the remote team to hear from the client why their business needed them and talk about how what they were doing made a real difference. It seems like a small thing, but you can get a lot of mileage out of it when it happens.

Some questions to ask your individual team members:

- Why is what we are doing important to the client/customer?
- What problem are we solving for them?
- How would we know if we solved the problem?

If the answer to any of these is "I don't know," then you have a problem that needs to be addressed.

The Best Functional Is Cross-functional

Global Tek, for the most part, had a cross-functional team before splitting off the remote team. By "cross-functional," we mean the team had all the skill sets and roles required to do the work represented in the people on the team. This is a big part of what made it successful in the first place. Cross-functional teams, while they at first seem more expensive, will get more work done, at higher quality, and faster (i.e., cheaper) than any other type of team. The empirical evidence backs this up and has been covered in a multitude of other books if you can't accept that on faith.

Anytime you break up a cross-functional team, you introduce at a minimum the overhead of handoffs. In most cases, you also add delays, decrease quality, degrade clarity, lower reaction time or the team's ability to pivot and limit innovation. By splitting off the remote people as dependent team members rather than an independent team, Global Tek lost most, if not all, the benefits of a cross-functional team.

At the risk of being redundant, everything we talked about up to this point applies. Find ways to keep the unscheduled communication going and ensure everyone understands the "why" as well as the "what," and it will go a long way toward forming one team out of two.

Outside of those things, the answer is very straightforward. Keep the team sitting together. If they can't physically sit together, then find ways to make it feel like they are. Utilize instant messaging, video conferencing and collaborative digital whiteboards. Make time for small talk, getting to know each other and nonwork conversations. Take the occasional workshop as an opportunity to get people physically in the same room. We've also found that it is important to give everyone the same handicap — if one team member is remote, then all team members have to join instant messaging and chat. Not doing this tends to ostracize the remote team member and significantly reduces their opportunity to communicate and therefore contribute.

If you're worried your team isn't functioning like a cross-functional team, there are some questions you can be asking to help push them in the right direction. Some of these will be familiar because they are pulled from other areas of this book.

- What is preventing work from flowing smoothly through the system? Are there blockers, pain points or missing skill sets that need to be addressed?
- What is preventing the teams from delivering direct business value?
- Are there any required skill sets not represented on the teams that own the work? If so, how do we build that skill set into the team or better coordinate with teams that already have that skill set?

Only Be Sorry Once

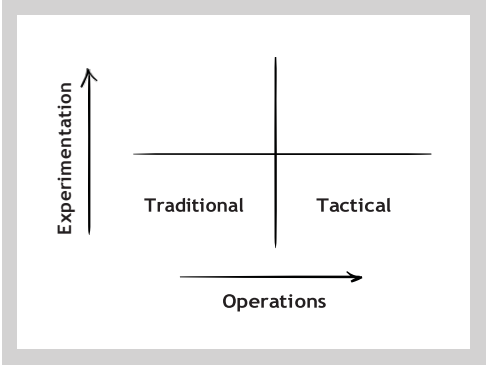
Okay, this one isn't really about Agile, but it's still important enough to mention here. We got into a bad place with Global Tek where we couldn't stop apologizing, and when we finally tried to stop, it was really painful because we had formed a new pattern in our

relationship with our clients. It seemed like the clients were getting what they wanted, but in reality, all the same issues were being magnified by the cycle. Even when things went okay, clients still felt the need to fish for the apology and the accompanying "free work."

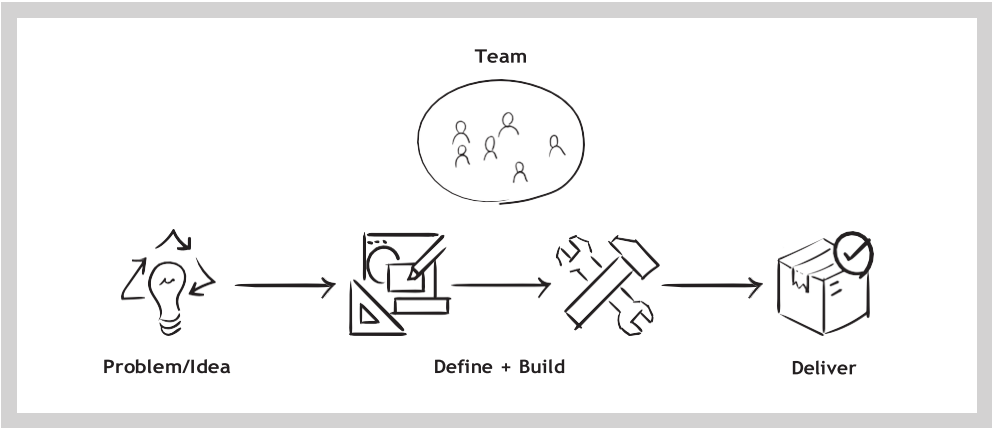
Only apologize once. After that, focus on prevention and future results. When things go wrong, as they did in this case, it is rarely just one side's fault. Your best bet is to set up a retrospective focusing on solutions for the problems, setting realistic (if painful) expectations, and use the iterative Agile approach to deliver a stream of value at a sustainable pace. If that's not enough, you don't want that client anyway — they will always find ways to cut into your margins, drive away your people and make your life miserable in general.

Overcoming Obstacles:

Comprehensive Documentation

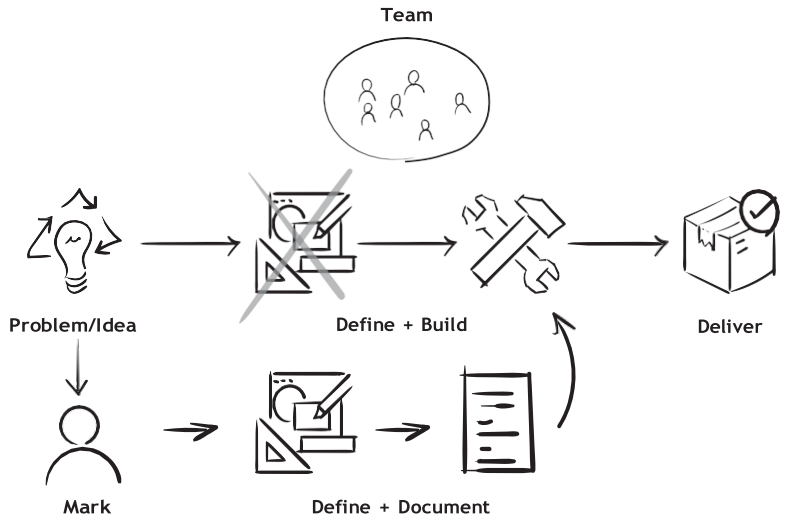


Erica, a project manager for Widgets, was handed a project to work on. Erica and her team had been using new ways of working for the last several projects, and they were just starting to feel like they were getting the hang of it. They were working together cross-functionally, meeting for standups every day, building a backlog and prioritizing work from it, along with a myriad of other Agile practices. In their last retrospective, the team recognized that several of their current struggles were due to not following the practices they knew were best. They expressed their desire to change and vowed to do things the "right" way. They were ready to be a self-organizing team and own their work.



The marketing manager for the project was Mark (remember him from the earlier parable), who was very eager to get things started — especially since the project had been waiting in the queue for six months. Mark made sure that the team had the 400-page requirements document that had been finalized during the project request process. As the team met for a project kickoff, they were deeply ~~concerned~~ impressed by the sheer amount of information that had been gathered. Digging deeper, they were increasingly more worried. Many architectural and design decisions had apparently been made that would significantly increase effort with only marginal, at best, increased value. When they raised their concerns to Erica, she informed them that the design phases for the project had already been completed. After all, with such a lengthy requirements doc, they must have known what they were talking about.

The team put their heads down and proceeded to develop the designed solution, albeit with severe reservations. A couple months into the project, the team had encountered countless situations for which the predefined specs must be ignored or changed to adapt to the way that the solution actually worked as opposed to how it theoretically could/should work. Few, if any, had fully read the 400-page requirements document, and as the project progressed, the document resembled reality less and less.



Here are a couple key principles:

- You Will Only Learn More From This Point On: An upfront requirements document was created at the point where the team knew the least about the solution. Plans need to be crafted so that they can be more defined as additional information is gathered.
- You Can Move a Wall With an Eraser — or a Sledgehammer: Locking in an error or a bad decision from an early phase results in every subsequent phase paying for the error/decision. If we reduce the time it takes to receive feedback on our decisions/choices, we gain the ability to rapidly learn from our mistakes in time to make meaningful changes.
- Documentation Is Like Water, but It Isn't a Product: You have to have some of it to live, but if you have too much, you'll drown. Documentation may represent value, but it is rarely something that customers are willing to pay for. Don't make the mistake of fooling yourself into believing that producing output is the same as achieving outcome.
- Don't Believe Everything You Read: Documentation gives the false assumption of accuracy, completeness and value. In addition, heavy documentation makes people feel like it has more value than it really does, and we become reluctant to change it. Making early documentation appear rough is a great way to appropriately highlight the incompleteness of early planning.

You Will Only Learn More From This Point On

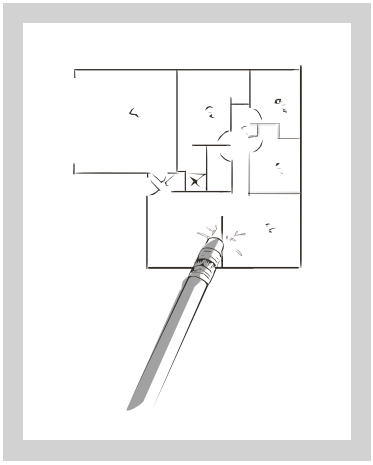
How much do you actually know about the solution to a problem at the start of the project? Mark prepared an extensive requirements document, and his effort was certainly valuable in helping him understand the current state. Unfortunately, his effort didn't result in learning for the whole team. Actively learning from hands-on efforts results in a deeper understanding of the details and a better grasp of the whole situation than you can get from reading a document. This is one of the reasons why handing off from one phase to another, or from one team to another, can be problematic.

Is the solution that we are building what was initially envisioned? If so, the team is likely not learning, or at least not adjusting, their direction as they learn. As a leader, you should fully expect plans to change and adapt. Sticking to a plan or documentation is often an indicator that the team isn't learning and that the end solution or outcome won't be as good as it could be.

How do you know that the solution you are building will actually work? Teams should be able to show that they have validated their assumptions throughout the project. That validation should start when they have rough concepts and continue as they launch to the public. While validation can come from many sources, the best validation is from real customers interacting in a natural environment. Focus groups are great, but live pilots are better.

"But we have to do it that way. That's what the documentation calls for." This phrase is all too common in product development and marketing organizations. We think that many view the plan or initial commitment as a promise and, if they adapt the plan, then they've broken their promise. Used correctly, adaptation serves to maximize the outcome in the face of a changing environment. Things may have changed since the start of a project to make the initial solution nonviable. In this situation, changing the solution allows you to avoid the inevitable failure. In many cases, it's not that the initial solution is nonviable, but rather that a better solution presented itself.

You Can Move a Wall With an Eraser — or a Sledgehammer



How often are you validating your assumptions and plans? (Yes, we know that this is a repeat — it's that important.) Validating a design when it is only pencil and paper, or when a wall has been framed but not finished, may not contain very many features, but it does provide the necessary platform for feedback. The team may be able to identify that, while the problem was interesting, and their idea was good, it just wasn't going to be a profitable endeavor.

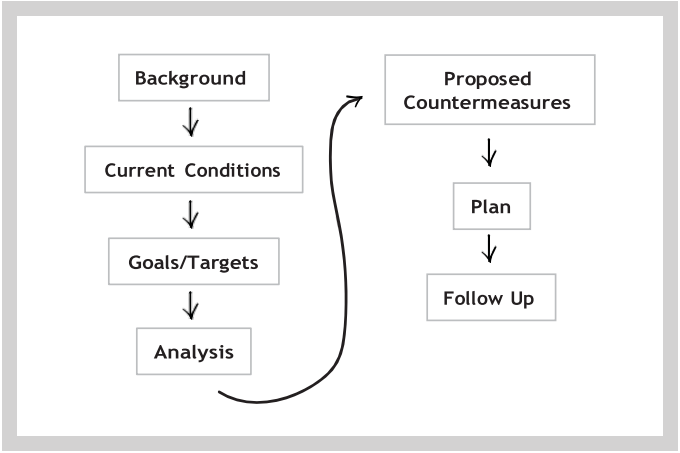
How can I start a project without a plan? Rather than attempting to lock down every little detail in a plan before starting the project, the team can meet for a project kickoff and set the context for the project. This includes a deep understanding of the problem to be addressed, high-level features that could provide value to customers and an initial plan for the first release. While not able to provide all of the value, or complete every feature, the team's initial plan can focus on providing a workable first release as quickly as possible.

Documentation Is Like Water, but It Isn't a Product

Can you describe your project/problem/product in one minute? On one page? The ability to concisely describe something increases as you have a better understanding of the thing. The more documentation required typically indicates that more investigation

is needed. It's not a bad idea to encourage your teams to maintain a one-page executive summary of the project. Personally, we're a bit partial to A3s[16] or Lean Canvas.[17]

A3s first originated at Toyota as a simple method for capturing information about a problem, analysis of data about the problem and a description of the solution. "Managing to Learn" by Jon Shook is the authoritative book on the subject and the intended flow of the A3 is described in the following diagram.



All of the content is intentionally limited to a single 11"x17" page (A3 size) to drive clarity and conciseness. Not only does this single document serve as the worksheet for the team solving the problem, it also fulfills the role of an executive summary. The Lean Canvas is a similar single-page document, but it is framed around entrepreneurship and business models rather than operational problems. Either approach dramatically reduces the volume of documentation while increasing the emphasis on valuable content.

Are you currently over-documenting? Can you show me a presentation of the current state of your project/product? Documentation created specifically to communicate outside the team (i.e., to people not doing the work) is often not valuable. The end customer doesn't pay for it. Let leaders/managers see the work that is being accomplished. Help them understand that a demonstration of the work in process, not a special presentation, is the best indication of the status of the project/product. This means that team members need to become comfortable with sharing work that may look like stick

figure drawings, Post-it Notes, or sketches on a whiteboard.

How much value do your teams place on documentation? Is a PRD, PSD, BRD, FPD (or some other three-letter acronymed document) considered a valuable project artifact or deliverable? Do you hand off the documentation? Any of these would indicate that there is likely too much value placed on documentation. Don't get us wrong; documentation does provide value, but it is really only valuable to the people doing the work. Even then, it is only valuable in moderate quantities and when it serves some purpose.

When was the last time your documentation was updated? Excessive documentation is difficult to keep up to date and isn't updated as part of the process of getting the work done. If the documentation isn't being kept up to date, then it may not be providing the necessary value to the people doing the work.

Don't Believe Everything You Read

Is there a request form to update documentation? Maybe there is a special process to edit, delete or update? If it takes a special process, then it is likely too much. As it is even more likely to not be updated. This means that when people actually access the documentation, they will think that they are getting accurate information, but in actuality are getting old data that doesn't reflect the current reality.

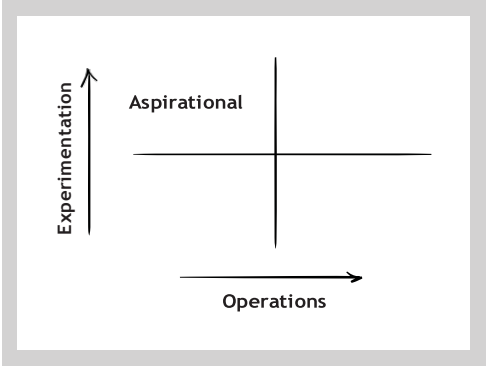
How detailed is your documentation? We know it sounds weird, but the more detailed the documentation, the more likely it is to be inaccurate. However, the more detail we see, the more accurate we assume it is. The easiest way to describe this is to compare it to a home scale. It has a couple of friendly digits after the decimal to measure weight in tenths and hundredths of a pound. Seeing this "accuracy," we decided to measure an item that only weighed a fraction of a pound. It was only then that we realized the scale was only capable of measuring to the fifth of a pound. Because of the level of precision, we had assumed a level of accuracy. The greater the amount in the documentation, the greater your assumption of accuracy. Don't be tricked.

Does your documentation reflect learning, or assumptions? Documentation is not the

end product, so it is very easy to say that something works in the documentation, but not to have it reflect reality. Our favorite repeated example of this is in software development when we're told that something is "done." Sure, it may have been documented as done, but is there actual proof that it has been completed? Does it actually work?

Overcoming Obstacles:

Loss of Control and Frustrated Team

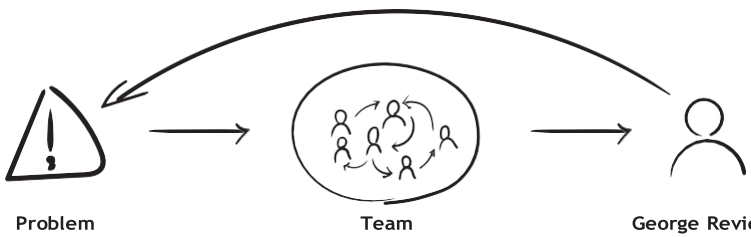


George was a creative executive at NewDesigns Inc. when its designers were in the middle of an Agile transformation. They were genuinely excited for the opportunity to work in a new way, and George was, in turn, encouraged. Throughout his career he had seen a number of different initiatives spin up and die, without resulting in the impact that had been promised. What he'd seen of Agile made him think it was better than other initiatives, but he wasn't holding his breath.

The designers jumped in and immediately set up self-organizing and cross-functional teams that could attack the problems and come up with solutions. Large problems were broken down into bite-sized problems and tracked visually with a Kanban board. Rather than wait for work to be assigned to them, each team member took the next prioritized

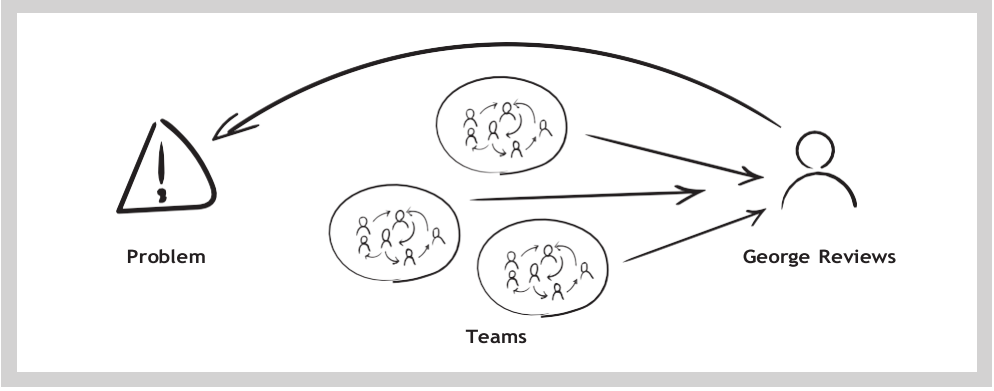
item from the backlog, even if it wasn't something that they were the expert at. This was done intentionally as a way to build broad capability within the team. While George wasn't comfortable with every way that the team self-organized, he tried to micromanage as little as possible.

At a minimum George required that he be involved in approving all designs, but the team was responsible for how they worked otherwise. Unfortunately, he wasn't always pleased by the designs that he was seeing. The designs just didn't "feel" right. As time went on, his unease continued to rise until he was convinced that the quality of design was decreasing. He was even more concerned when he realized how many design work sessions were happening that he wasn't invited to.



It wasn't too long before the team realized that their client wasn't nearly as excited for the new way of working as they were. The client still required mountains of documentation for requirements and copious notes whenever it looked like the scope was going to change. When the team escalated the issue to George, he told them that they couldn't control the client and they needed to do whatever the client asked. After all, the client was the one paying the bills.

In an effort to guarantee quality, George instigated regular internal review and approval meetings, and also required his attendance in all design workshops. Over the next several weeks, George saw that his changes were having an impact. He felt much more confident and comfortable with the outputs. Unfortunately, he also felt exhausted. He just didn't know how he was going to keep up with all the work.



What he wasn't paying much attention to was his people. They were steadily becoming more and more disgruntled. He didn't think much of it when the first couple designers quit, but after the third and fourth, he was concerned. What was happening? They had introduced a new way of working. It was supposed to improve employee retention and empowerment. They were working in cross-functional teams, weren't they? What could he do differently?

Here are a couple key principles:

- Progress Is More Important Than Standing Still: Because life is like skating up an ice mountain — there is no such thing as standing still. Letting your people know there is progress grants hope.
- Manage the People, Not the Work: Manage the people so that they can do excellent work. Trying to manage the work will disempower teams and be too much for a single leader to actually control. In addition, when the leader manages the work, the people don't learn how to do it themselves. It's a vicious cycle.
- Manage the System, Not the Work: The true role of leadership in an Agile environment is to create and foster a system that encourages excellent work.

Progress Is More Important Than Standing Still

Are obstacles for the team being removed? Do you make it your priority to remove obstacles? As a leader, that is one of your primary responsibilities. It isn't to review and approve. It isn't to punish, praise or discipline. It is to remove the things that make it difficult for your people to do their jobs.

Are regular updates on progress being provided? You may not be able to force another company, group or department to change, but you can work with them to understand the situation. In this case, George could have shown that he was working with the client to make changes. Progress inspires hope. Think about your experience at Disneyland. What would happen if you had to stand still and wait for your chance to take a ride? Would you wait? There is a reason why waiting in lines works. It's because we can see the progress we are making and it gives us hope.

How long do obstacles remain before being removed? Even if you are making progress, if it takes too long the team will eventually give up. We recommend setting a time limit for obstacles and an escalation path if it is blocking the team for more than a certain number of days.

Manage the People, Not the Work

How much of a leader's time is coaching and leading people? George spent little, if any, time with his people; instead, he focused on the work. He was quickly overwhelmed by the demands on his time. If he continues in this way, it will lead to exhaustion and burnout.

Do you see your responsibility in leadership as one of approving? As a leader, you don't have enough time to know every little detail. And, if you don't know every little detail, should you be the one approving? Leaders can provide guidance and coaching, but those doing the work typically know more about the problem at hand and are in a better position to know if the solution is good or not.

Are you trusting your people to do excellent work? If you hire good people, you need to trust them enough to continue to do good work. By managing the individual, you empower them to take responsibility for their work. You'll also end up focusing on how they can get better overall rather than critiquing every little piece of work output.

Manage the System, Not the Work

Is there a defined system for your people to work within? You'd be surprised by how often there isn't a system, or at least not a well-defined system, for getting the work done. We've seen countless workplaces where the outputs and deliverables are overly defined, but the way that the work is accomplished is left up to every individual. You might think that this is empowering, but it is not. Often the reverse is true. Each individual spends so much time trying to define their own system that they don't have the time or energy remaining to focus on being creative — on actually creating value. This is incredibly disheartening and leads to poor outcomes as well as disgruntled employees.

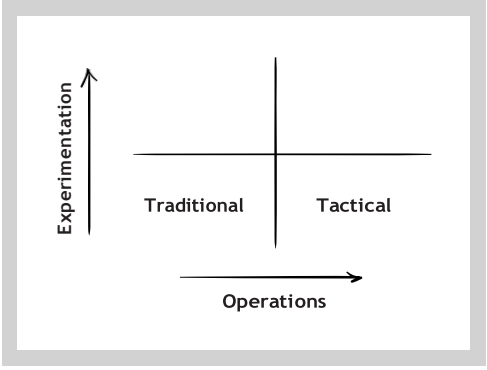
Does the work environment and system set up your people for success? George did notice that the system in place wasn't yielding the best outcomes. Unfortunately, his remedy for the situation was to insert himself and directly manage the work rather than just fixing the system. Take a careful look at the system that your people work in. Is it

setting them up for success? Or for failure? If it is setting them up for failure, then you need to take responsibility to address it. That doesn't mean that you have to make all of the changes yourself. You just have to take ownership of the system. It will generally work out best if you involve your people in improving the system instead of just changing it all by yourself.

Do you blame the people or the system when failures inevitably occur? Failures and errors will always occur. You won't be able to completely avoid them. When they occur, remember that the system allowed the failure to occur. Focusing on fixing the system is much more productive than berating individuals.

Overcoming Obstacles:

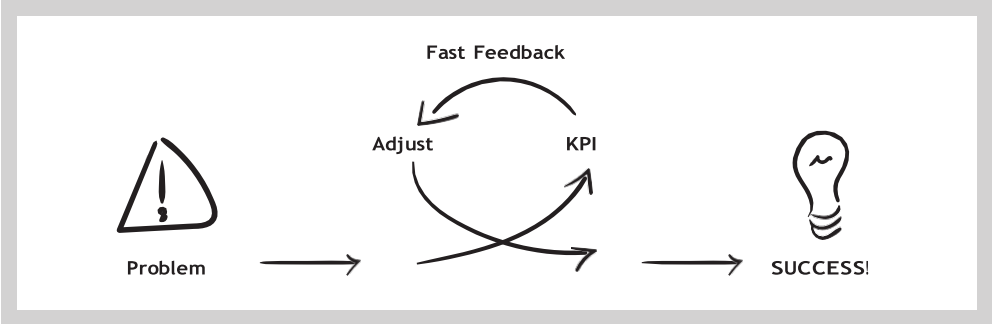
Partnership Accountability & Success



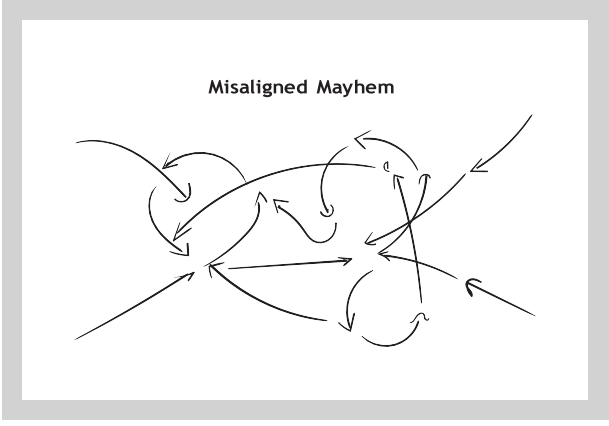
Momentum LLC had recently transitioned to an Agile way of working. It was an abrupt, but not bad, transition, and they were trying to get everyone comfortable with the new way of working while still plowing full steam ahead. It was an all-around tough situation as not only were they changing the way people worked, but they were also facing imposing business constraints resulting in a huge impact being required in a very short time frame. This is probably a familiar situation by now.

The best thing Momentum had going for them was that they knew, without a doubt, what success looks like. They knew exactly what KPIs they had to move and by how much. They had clear ways to measure the impacts their efforts were having. They could even test changes with small portions of their user base to ensure they were effective and

functional before making the change universally available. This was probably one of the best feedback loops we've seen in a long time.



To move forward quickly, Momentum brought in multiple outside partners to help with the Agile transition, but, with few exceptions, trust never really grew in the partnership. The various relationships were typified by blaming, finger-pointing and not-my-jobbing. Each group in the partnership had their own vision of what Agile needed to be in this situation that didn't really match anyone else's. The lack of clarity and buy-in around the approach went a long way in preventing the real and lasting change that was needed to support Momentum's vision. The initial transition had been accompanied by a clear message: How they had worked in the past wasn't enough anymore. Unfortunately, confusion about vision allowed older bad habits to live on and eventually make a resurgence.



As time went on, the resurgence of bad habits continued and people began to point at the new ways of working as "the problem." Leaders stepped in to fix the problems and set clear processes for the teams. The leaders, however, were faced with the same problem as the teams. They didn't have a cohesive vision and began responding to whatever was causing the most pain at the moment. Eventually, with enough complaints about the new ways of working, leadership determined to revert to the old ways since they were less confusing.

Here are a couple key principles:

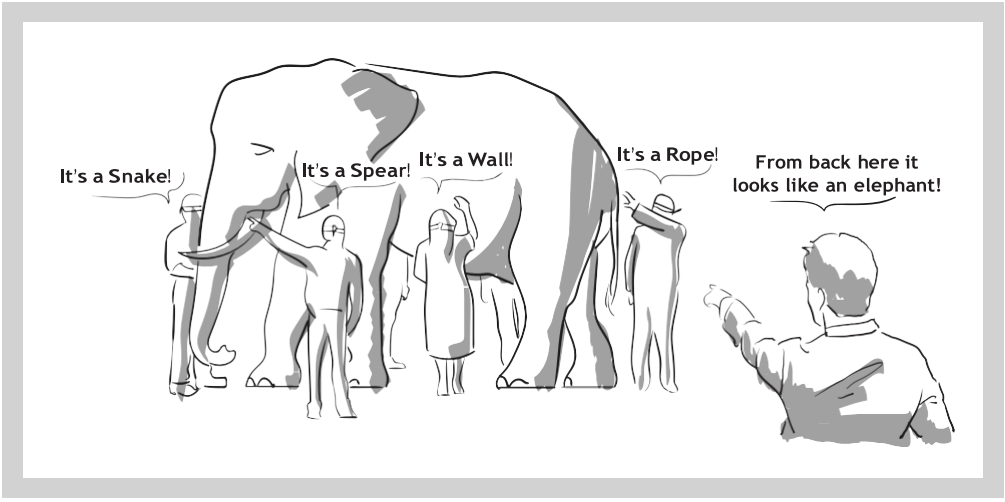
- You Need a Compass and a Destination: KPIs, early releases, and other customer feedback loops are your compass, and without them you are likely to sink. Without a destination, or a vision, the compass won't help you get anywhere.
- If You Don't Trust Your Partner, You Don't Have One: To move and pivot fast, you have to have strong partnerships built on trust. If trust isn't there, neither is the partnership.
- You Can't Fix People's Problems for Them: Empowered and accountable teams guided by clear and strong principles, values, and a clear vision are required in today's Agile world. Fixing problems for people doesn't equip them to fix other problems in the future.

You Need a Compass and a Destination

Momentum had something that surprisingly few organizations have: clear metrics. They knew how their actions impacted the metrics, and many of their decisions were data-driven. Normally we would place a company like Momentum on a pedestal to show others what it takes to be world-class. However, without a single, cohesive vision, excellent metrics had nowhere to go. While implementing change cold turkey can be painful for an organization, it often leads to better transformations with significantly less backsliding than you find when organizations attempt to ease into it.

There needed to be a clear vision of what a transformed team and organization

looked like. We often say that vision can't be done by committee because that results in a fragmented perspective. Rather, you need to have an individual who holds the vision and communicates the vision constantly. The vision holder can't be expected to come up with the vision by themselves though. That's just too much to ask of a single person. They need to gather ideas, feedback and expert advice along the way.



One effective method of conveying a vision is to state it in terms of principles and values. Stating what you value, and giving a real-life example, allows others to begin placing themselves in a new paradigm. One of the reasons we like Agile is because the Agile Manifesto does this so well. We begin to see a vision of the future when we hear, "We value individuals and interactions over processes and tools." Stating this value doesn't mandate a particular solution or tactic, but does highlight what direction we are going. If we are changing, and find that our practices no longer align with the stated value, then it is time for us to rethink our current path.

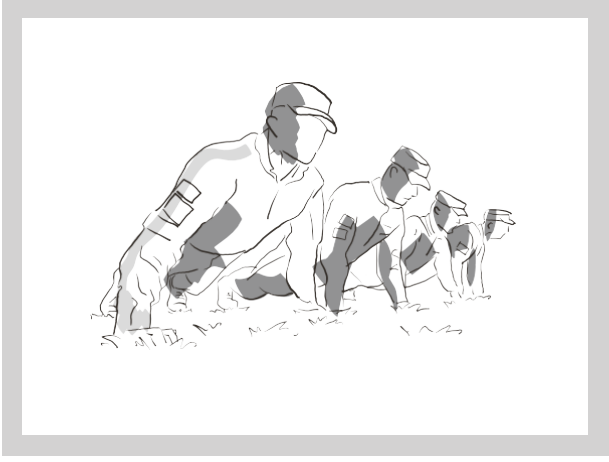
If You Don't Trust Your Partner, You Don't Have One

Bringing in outside partners to help with something that you aren't experienced with is often a great approach. Those partners have already seen many of the common pitfalls and can help you avoid making the same mistakes. Unfortunately, with Momentum, a

true partnership and sharing of knowledge never grew and many potential benefits of the external partners weren't realized. It's pretty easy to see that there were no partners in this endeavor, just companies and individuals attempting to work for their own benefit. This is by no means an isolated incident, and we have witnessed similar situations in many engagements.

As humans, we want to belong. Daniel Coyle has some great examples of this in his book "The Culture Code." The trick is getting people to feel like they belong in a way that is productive. This case shows that the feeling of belonging to a particular company was stronger than the feeling of belonging to a team. For instance, when you asked someone what they did for work, they would likely say something like, "I'm a creative director for Agency Y." That statement suggests that the individual identifies more as a creative director at Agency Y than they do as a member of the mobile app team.

When we stand up cross-company teams, we often begin by doing introductions and forbidding team members to say which company they work for, or even what title they have. Instead we ask them to tell us about a hobby, something they did over the weekend, and the types of things that they love to do at work. Subtle social cues like this go a long way in breaking down interpersonal barriers and building up teams. It reinforces the fact that the individual and their talents is what makes up the team, not a title, not a company.

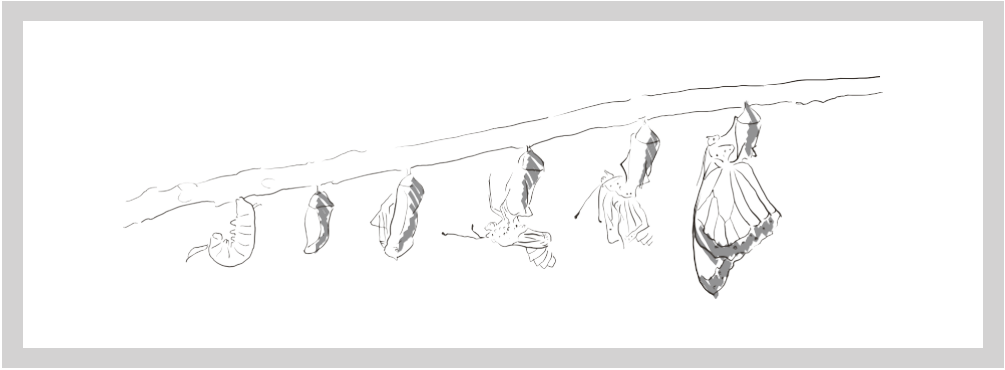


Another way to begin integrating a partnership is to give every team member joint responsibility. The military often does this as a way to build team cohesion. If any member of the platoon causes an infraction, the whole platoon is required to do push-ups. In the short term, this may cause animosity toward the team member who caused the infraction, but teams quickly realize that animosity doesn't make the push-ups go away. The team figures out a way to avoid infractions. This approach ignores finger-pointing. It changes the team member's perspective from "yours" and "mine" to "ours."

You Can't Fix People's Problems for Them

Many leaders, including us, have made names for themselves by fixing problems. We think like engineers. We see a problem and immediately begin thinking of ways to fix it. Over the course of our marriages, our respective partners have slowly introduced us to the perspective that not every problem can, or should, be fixed by you. It did take years, and many conversations to pound it into our thick skulls, but we finally got it. People grow by finding and fixing problems in themselves and in the world around them. If we fix the problems for them, then they don't have the opportunity to grow.

A favorite example of this is the butterfly. When emerging from the cocoon, the butterfly faces the greatest struggle of their life. The former caterpillar must break out of its protective shell after not eating for possibly weeks. Then they expand their wings and stretch muscles that didn't even exist before the cocoon. It seems like a horrible trial, and one might be tempted to step in and help out. However, the very trial of emerging prepares the butterfly to f ly, and were you to assist them, they would never achieve their potential.



The new role of leadership isn't to solve the problems, but rather to create an environment where problems can be solved. The leaders needed to set a vision and guiding principles before empowering the teams to solve their own problems. With the vision providing a direction and a view into the future, with principles guiding actions, the teams can quickly see where their solutions may or may not align. Leadership also needs to stick to the vision and principles through the hard times that will inevitably come. Life is full of surprises, and when we are attempting to transform, things often get worse before they get better. It takes courage to stick with a vision when things start to get difficult. Again, think of the butterfly. For a time the butterfly is in a worse situation than they were as a caterpillar. From outside, the cocoon seems to be static, but once some key metamorphosis is accomplished, we can start to see progress.

The Moral of the Story

Too much of the Agile world has become codified and inflexible. It's the opposite of what agility should be, and it's come about in many ways due to the expansion of the certification economy and the stake certified coaches have in its continuation. We hope that if you take nothing else away from this book, you have seen that Agile is not a binary state. Agility involves a broad spectrum of activities, rituals and behaviors that are designed based on some key principles. When you look at transforming your organization and building a competency around agility, don't be afraid to invent, change, and grow Agile practices as part of your journey.

Remember, it only takes four steps to be Agile.

1. **Decide where you want to go.**
2. **Take a small step toward your goal.**
3. **Adjust based on what you learned.**
4. **Repeat.**

This same inflexibility also leads to many questioning the validity of Agile in industries outside of technology, or even more specifically, software. We couldn't agree less. True agility leads to increased value in nearly any situation, whether that be in the workplace or the home. Yes, we apply the same principles in our home life. We are those people.

If nothing else, reeducating your organization to be principle-based is truly empowering and will unlock previously hidden levels of performance. This is the essence of cutting through the bureaucracy. Agile principles are some of the best at unlocking this potential, but don't hesitate to stretch into Lean and other areas as you see fit. Find the principles that work for your situation, industry, and people and use them. Show your teams and leaders how principles should guide the selections of tools, processes, and even features.

Become an evangelist not only of your principles, but also of the potential. Don't hesitate to point to examples of other organizations or competitors who have doubled their productivity, cut their time to market in half, or drastically increased their quality using an Agile approach. Just don't assume what worked for them will work for you. There is no silver bullet.

So where do you go from here?

1 All Change Is Hard

Go into this with your eyes open and expect resistance. Organizations have cultural and procedural inertia that you will have to overcome. Even worse, following any change, you will first see a decrease in performance before seeing the true benefits. This is the "j-curve" of change management, and we see too many organizations stop at the bottom, forming a new and worse normal, before they ever have a chance to see the peak. Be prepared to weather the storm.

If it's too easy, then something is wrong. When you start down the path of transformation and everything seems to be flowing smoothly, there is a good chance this means you aren't really changing. Either you're leaving money on the table in terms of the

gains you could be getting or the bureaucracy is hiding the reality. Expect and embrace the pain since we all know "no pain, no gain."

2 Solve Real Problems

Often, transformation takes the form of new processes and procedures broadly being applied across an organization. We see this often when an organization decides it's going Scrum, and just as often we see this fail. This type of approach can feel disingenuous and does nothing to overcome resistance.

Instead, find your organization's more painful problems and bring your new principles to bear in solving them. Not only will this approach help solve real problems, but, at the same time, it will show your dedication to the new way while also demonstrating its value. Nothing will more deeply embed the new mindsets and tools within your organization.

3 You Get What You Measure

We haven't talked a lot about metrics in this book, but we cannot stress enough that they are a double-edged sword. You truly do get what you measure, and teams will naturally game the metrics. It's not malice; it's human nature to optimize based on how you're evaluated. So use metrics for good (value), not evil (performance).

Remind yourself and your organization at large that if your teams are delivering value, the specifics of how they are doing it don't matter all that much.

Also, if you focus on the value to your customer, everyone feels better about the work. Customers are happier with the results. Teams feel more engaged and rewarded for their work. You get to spend less time worrying about velocity and all that crud, and more time staying ahead of your competitors. It really is a win-win-win situation.

We aren't saying performance metrics aren't important, by the way. They are an important part of a team's ability to continuously improve, and teams should still measure their own performance metrics. But teams should be evaluated on the value

they create, not the number of story points they complete.

4 Get Help

We know, we know, we know ... We don't seem very pro-coach right now. But in reality, for almost any successful transformation, you need a coach. We would recommend that, when looking for a coach, you look more at their flexibility and breadth of knowledge than their certifications. You want someone who can pull from a wide selection of principles, systems and tools to help you build the approach you need over time.

Coaches will provide you with an invaluable third-party perspective that individuals in the day-to-day work, no matter how talented, just can't provide. You will, at times, hate them for it, but having that timely discussion with someone who is focused on the ways of working rather than the work itself is a lifesaver. It's the only way we have found to consistently right the course when the going gets rough.

5 You Are Never Done

An Agile transformation journey does not have an end state. There is no magical time when you have fully transformed and are completely Agile. This is more of an evolution than a transformation, and when you stop evolving you become a dinosaur quicker than most believe possible.

So, to leave you with our closing thought: never stop. What got you here won't get you there. What was successful yesterday may not, and probably won't, be successful tomorrow. The heart of Agile, more than anything else, is continuous improvement. Stop improving, evolving and growing at your own peril.

References

1. https://docs.broadcom.com/docs/the-impact-of-agile-quantified
2. http://justinrohrman.com/blog/wp-content/uploads/2016/03/ImpactofAgileQuantified2015.pdf
3. https://www.businesswire.com/news/home/20080804005495/en/New-Study-Shows-That-Agile-Teams-Are-37-Percent-Faster-to-Market-and-16-Percent-More-Productive
4. https://www.bcg.com/publications/2019/agile-works-measuring-impact
5. Patterson, Grenny, McMillan, Switzler *Crucial Conversations: Tools for Talking When Stakes Are High*. McGraw-Hill Education; 2nd Edition (September 7, 2011)
6. Jonathan Raymond. *Good Authority: How to Become the Leader Your Team Is Waiting For* Ideapress Publishing (October 4, 2016)
7. Kim Scott *Radical Candor: Fully Revised & Updated Edition: Be a Kick-Ass Boss Without Losing Your Humanity* St. Martin's Press; Revised, Updated ed. Edition (October 1, 2019)
8. https://agilemanifesto.org/
9. https://en.wikipedia.org/wiki/Little%27s_law
10. https://www.adpri.org/assets/the-global-study-of-engagement/
11. https://www.scaledagileframework.com/
12. https://blog.crisp.se/2014/03/27/henrikkniberg/spotify-engineering-culture-part-1

13. https://en.wikipedia.org/wiki/Cargo_cult
14. https://en.wikipedia.org/wiki/Five_whys
15. https://glossary.ametsoc.org/wiki/Persistence_forecast
16. Shook, John, *Managing to Learn: Using the A3 management process to solve problems, gain agreement, mentor, and lead*, Lean Enterprise Institute, 2008
17. https://leanstack.com/lean-canvas
18. Coyle, Daniel, *The Culture Code: The Secrets of Highly Successful Groups*, Bantam (January 30, 2018)

Figure 1. https://www.leadingagile.com/2016/09/reclaiming-agile/

Figure 2. Kniberg, Henrik, Product Ownership in a Nutshell
https://www.youtube.com/watch?v=502ILHjX9EE

Figure 3. https://shingo.org/shingo-model/