LEADINGAGILE

AGILE TRANSFORMATION

MIKE COTTMEYER





INTRODUCTION

FOR YEARS, WE'VE TREATED AGILE LIKE IT'S SIMPLY A PROCESS. Just a new way of doing what we've always done. Why is it then, that when we train people on how to use the process, they so frequently struggle to see any benefit? Scrum can be described in less than 20 pages. SAFe can be taught in less than a week. Why is applying these concepts so difficult for so many organizations seeking to get the benefits of going Agile? Why is it so hard…what could we be missing?

The answer is that much of what makes Agile work, what makes it really work, was lost as the general population of programmers and project managers rushed to take advantage of this new way of working. Agile processes require a certain context to be effective. They require us to form teams, build backlogs, measure work, and control work in a certain way. It's more than just the roles, ceremonies, and artifacts of Scrum. It's the ecosystem Scrum operates in that really matters.

The problem is that this ecosystem doesn't actually exist in most of the companies that are actually trying to adopt Agile—at any kind of scale. Sure, many companies can adopt Agile on a small, self-contained project, but when it comes to changing how money is spent, how projects are approved, how return on investment is realized…we fall short. We struggle with forming the right kinds of teams, building the right kinds of backlogs, and measuring the right kind of progress. It seems everything in the organization is

> **AGILE PROCESSES REQUIRE A CERTAIN CONTEXT TO BE EFFECTIVE. THEY REQUIRE US TO FORM TEAMS, BUILD BACKLOGS, MEASURE WORK, AND CONTROL WORK IN A CERTAIN WAY.**

working against us making the kinds of changes necessary to benefit from a more Agile way of working.

The trick isn't to just teach people Agile, we have to find a way to systematically overcome the structural, procedural, and cultural barriers that are continuously getting in their way. That problem can either be solved top-down, bottom-up, or someway in between. In our experience we've found that removing the impediments will require executive support, dollars, and time. You'll need engagement from the senior leadership, middle management, and the people on the ground doing the work.

Furthermore, Transforming your organization isn't trivial...and it certainly isn't easy. You're going to need a plan. Yes. A plan. You're going to need a way to measure progress, demonstrate results, and to economically justify your investment. You'll need a way to establish hypotheses, validate and test those hypotheses, and pivot when things don't go as you might expect. You'll need a way to keep everyone on-board and engaged throughout the process.

> **YOU'RE GOING TO NEED A WAY TO MEASURE PROGRESS, DEMONSTRATE RESULTS, AND TO ECONOMICALLY JUSTIFY YOUR INVESTMENT.**

PLANNING YOUR TRANSFORMATION

This paper explores the thinking tools necessary to run a structured disciplined Agile Transformation. It's broken up into four major sections.

WHY

The first section explores the economic rationale behind the Transformation, and the nature of the assumptions we're making about what needs to change. This section consists of two sub-sections titled Business Case and Transformation Hypothesis. Business Case explores the common economic drivers leading most companies to invest in adopting Agile. Transformation Hypothesis explores the considerations that will drive your change approach and why one Agile Transformation may look quite different from another.



WHAT

The second section explores the thinking tools necessary to build your Transformation Strategy. It will also explore the basic patterns for what Agile looks like, both in the small and at the enterprise scale. This section consists of two sub-sections titled Theory & Approach and Reference Architecture. Theory & Approach will explore the

fundamentals of an Agile ecosystem, the patterns of scale, and how you get from one place to another when faced with competing business needs. Reference Architecture will look at the core patterns of enterprise Agility and offer a minimum subset of organizational patterns, governance models, and metrics necessary to establish an effective Agile ecosystem.



HOW

The third section explores how to actually orchestrate change and measure progress. This section is broken down into two sub-sections titled Change Model and Results Management. The Change Model section explores engaging your leadership team, and collaboratively building a shared understanding of the desired end-state, and how the organization will take steps to move in that direction. The Results Management section will explore how to measure progress toward the Transformation, and how to know if the Transformation is yielding the business benefit necessary to economically justify the change.



WHO

The fourth, and final, section will explore what roles are necessary to orchestrate the change as well as the skills and experiences necessary to effectively lead an Agile Transformation. This section is also broken into two sub-sections: Roles & Responsibilities and Skills & Experiences. Roles & Responsibilities will explore the outcomes and activities necessary to lead and implement an Agile Transformation; while the Skills & Experiences section goes into detail about the attributes of individuals in junior roles, senior roles, and executive roles on a Transformation initiative.



THE PAPER, THE TALK, & THE PRESENTATION

You'll notice that this white paper is color coded and full of images and icons. This paper is designed to work in conjunction with the talk that Mike Cottmeyer delivered at Agile2018...both in video form, and in Powerpoint. The colors corresponding to the Why, What, How, and Who sections are consistent between all three mediums, as are the icons and images. The idea is that you watch the talk and use the deck and this paper as a point of reference and a source for additional information, respectively. We did what we could to connect it all together and make it easy on you.

WHY

BUSINESS CASE

TRANSFORMATION HYPOTHESIS

WHAT

HOW

WHO

WHY

Before we get started with our Agile Transformation, it is important that we understand why we are making these investments and what we hope to gain from our efforts. Furthermore, we have to understand what changes we intent to make, and have a well formed point of view around how these changes are going to lead the business outcomes are seek to achieve.



THE BUSINESS CASE FOR AGILE

A business case will allow us to clearly state what our objectives are for the Transformation and give us a way to measure if we are successful.

It's Never About Agile

Adopting Agile is never about adopting an Agile methodology, it's always about delivering better business outcomes. Agile process is—at best—a means to an end. At worst, it can be a distraction that results in focusing on the wrong changes for the enterprise. Understanding your business goals will align your organization around common outcomes, help inform your Transformation strategy, and identify necessary tradeoffs. It'll also help identify if the Transformation is headed in the right direction.

Some of the common goals of going Agile are listed:



PREDICTABILITY

Agile tends to focus on adaptability as a key driver, but one of the most frequent stated goals of an Agile Transformation is predictability. Predictability means that we can reliably make and meet commitments to our customers. Predictability builds trust with our internal stakeholders, our customers, and our markets.



QUALITY

As organizations scale, it's common for quality to suffer, and the way in which it suffers can come in many forms. Sometimes we're missing features and functionality. Sometimes it's extrinsic quality problems in the form of defects. Other times, it's intrinsic quality in the form of technical debt. Quality issues erode trust with our customers and make our software difficult to manage.



COST SAVINGS

Most companies adopt Agile because they believe that it's more efficient and that it will reduce costs. But, their usually so overloaded with work that the cost savings are often difficult to achieve. What is achievable, is greater assurance that your people are focused on the problems whose solutions have the highest value.



EARLY RETURN ON INVESTMENT

Many organizations are struggling with long delivery cycles that make it very difficult to get feedback from customers. It also puts a burden on the organization in the form of long investment cycles before revenue can be realized. Early ROI means that we can put product into market in smaller increments to begin charging for the product and realizing revenue

.



PRODUCT FIT

One of the common goals of adopting Agile is making sure that we're building the right product for our customers. Agile gives us the opportunity to deliver in smaller batches, get frequent customer feedback, and change direction when we learn new things about our customers and their requirements.



INNOVATION

We know that well-formed teams, operating in the right market and in the right organizational context, can take advantage of Agile methodologies to exploit uncertainty. They'll be able to test product hypothesis, assess customer demand, and are free to explore what works.

It's not uncommon for organizations to desire all of these outcomes at the same time. While that's fair, sometimes these business drivers compete with one another to pull the Transformation in different directions. You'll also find that these drivers can often require different organizational structures and governance models that don't exist—or can't exist in the current company configuration. Regardless, understanding what success looks like will drive the right conversations as you begin to form your Transformation Hypothesis.



TRANSFORMATION HYPOTHESIS

A Transformation Hypothesis forms our basis of understanding for how we intend to approach the Transformation, why we intend to take this approach, and why we think that this approach will yield the outcomes we expect.

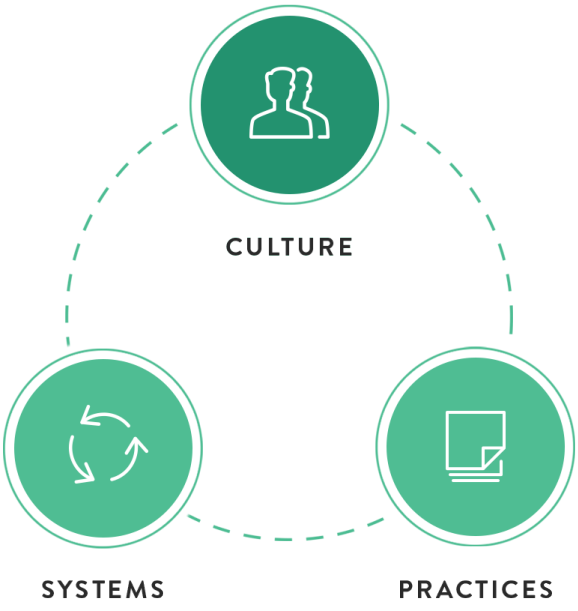
What Are We Changing?

There's a troubling narrative in the Agile community that adopting Agile, or orchestrating an Agile Transformation, begins with a cultural change. This is a tempting narrative because the resistance we experience feels like resistance to change. If people resist change, it's tempting to think that we have a culture that is resistant to change. While that could be the case, often times there are real constraints that make change difficult. If we want people to be open to change, we have to remove the impediments that make it difficult to change.

PRODUCT FIT

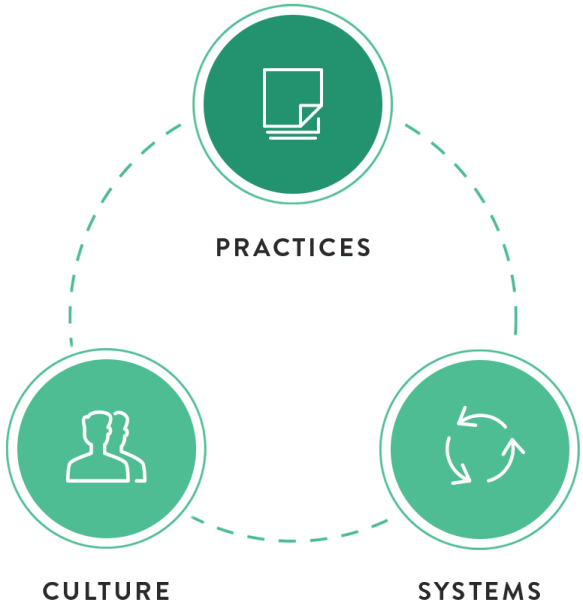
If we assert that culture has to change first, we're making the assumption that changing culture will lead to the necessary process and system changes necessary to sustain Agile in the enterprise. More often than not, people go down the path of cultural change, get excited about the possibility of Agile, and come back to deep-rooted process and organization issues that make Agile nearly impossible to implement in an effective manner. This often leads to cynicism, disillusionment, and people wondering why their peers, "just don't get it."

The problem is that a mindset change without proper guidance on how to form cross-functional teams, build refined backlogs, properly identify what needs to be measured and controlled, and actually implement those changes—often result in people simply "going though the motions" of Agile. This is a form of Agile where people are saying the right things, and maybe even doing the right things, but not achieving the results that they desire. This culture-first mindset is eroding confidence in both Agile and Agile methodologies.



PROCESS CHANGE

Often, Agile is implemented as a process change. This happens when you have people go through training and then optionally give them support from coaches to help sustain the new things that they've learned. A process-first Transformation is predicated on the assumption that adopting team level practices will yield—through iteration and retrospective—an understanding of the impediments that are preventing true Agility. Furthermore, we must assume that the team will be able to identify the root causes and actually resolve them.



The challenge with this approach is that many of the structural, organizational, technical, and governance issues are beyond the purview of a single team. While the team may see the impediment, it's unlikely they have the agency to actually do anything about the impediments they encounter. Many impediments at this level require executive support, funding, and time to remove. In the interim, simple process adherence results in our teams going through the motions of Agile without deriving any of the real benefit.

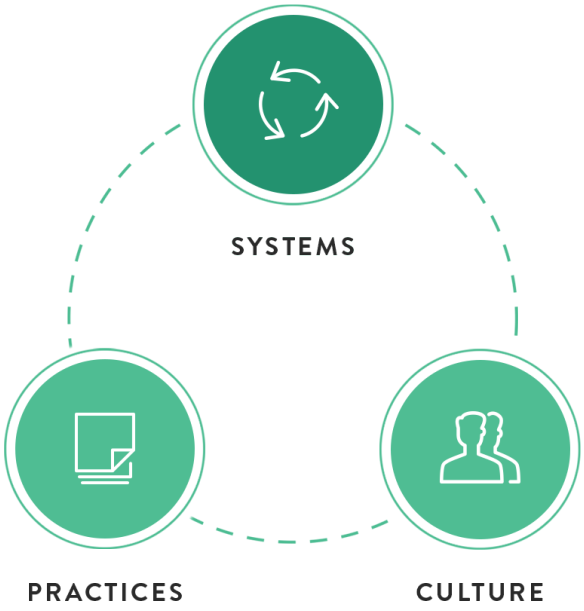
SYSTEMS CHANGE

Agile processes and Agile culture require a certain kind of ecosystem to effectively yield the promised business benefit. Agile necessitates a specific team formation strategy and a clearly articulated strategy for what we're going to form those teams around. It necessitates a specific strategy for how requirements are defined, how those teams will process the requirements, and how they will deliver them to their customers. Agile necessitates that we measure progress a certain way. Without the right fundamental ecosystem in place, and the right system of delivery, no amount of process education and cultural indoctrination will solve the problem.

Failing to install the appropriate environment for Agile to thrive will work against everything you do in terms of cultural indoctrination or through the introduction of process.

This is your first big decision as a change agent…as an Agile Transformation Leader. If you start with culture, you're betting that changes in attitude will drive changes to systems and practices. If you start with practices, you're making a bet that those practices will reveal issues that teams will be able to solve independently. The LeadingAgile approach requires that we define the ecosystem first, enable the system with practices that are respectful and aware of constraints, and then support and reinforce culture change over time. We believe this is the safest and most pragmatic bet that will increase your changes of having near term success with Agile.

> **ALIGNMENT ACROSS THE ORGANIZATIONAL DESIGN AND STRUCTURE, ITS ENABLING PRACTICES, AND ULTIMATELY ITS CULTURE IS WHAT LEADS TO LONG-TERM SUSTAINABLE TRANSFORMATION**



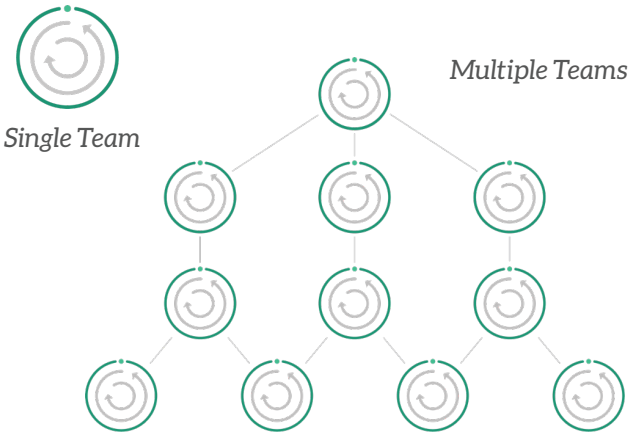
Things to Consider

That said, this isn't easy. Expect an uphill climb. There's a reason that many organizations start with culture, or start with process. We want to believe that if we simply point people in the right direction, they'll do the right things. Unfortunately, this isn't always the case and the momentum of the company often gets in the way. As an Agile change agent, you need to identify the constraints in the system and proactively work to overcome them. Sure, Agile is about inspecting and adapting, but—for the most part— 80-90% of the things that will get in our way are knowable up front. It's a good strategy to be upfront about dealing with issues we know we'll encounter.

SIZE OF THE ORGANIZATION

Size does matter. The strategy that you use for Transforming a single team is different from the strategy you'd want to use for a Fortune 100 company…or even a mid-size company of 300

product developers. If you're leading a group of six to eight people, sending people to culture school or Scrum school, along with a little coaching, might be sufficient. If you're Transforming 300 people working across an integrated product suite…you'll need to approach your Transformation with a greater degree of intentionality. If you're Transforming 12,000 people in a Fortune 100 company, the level of structure, planning, and coordination you'll need can be overwhelming. Knowing that these types of Transformations are different is a key insight necessary for crafting your plan.



RESISTANCE TO CHANGE

Resistance to change comes in lots of shapes and sizes. When most people think of resistance, they think of managers that don't want to change. People that are stuck in old ways of doing things and have no interested in learning or getting better. More often than not, your leadership wants to adopt agile and realize all the benefits of doing so. They might not know how, they might misstep, but the desire and intention is there. More than likely, you'll have the support of your senior leadership team, and maybe even the support of your execution team, but middle managers will get in the way because they don't know where they fit in this new world.

It's important that we understand resistance to change in a very human way. Quite often we have leaders that have been highly successful doing what they've been doing for a long time. They make a lot of money, they have mortgages and kids in college. Change… especially change that is big and scary and could possibly fail… threatens not only our job, but potentially our livelihood and the lives of our families. Being sensitive to the human beings involved in the change and making sure that everyone is informed and safe is a key component of leading change. If we don't create this kind of safety, we are likely to create detractors when we really need to have supporters.

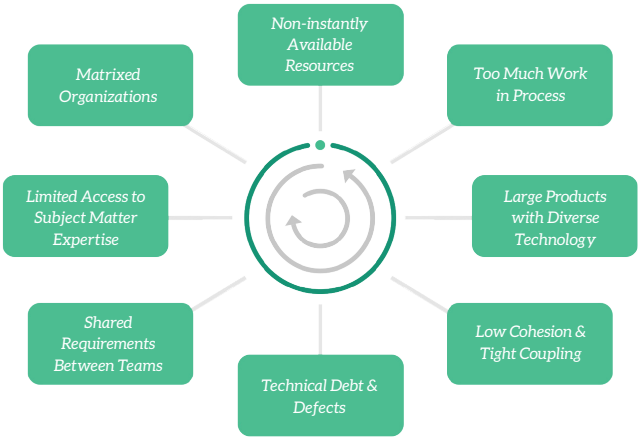


DEPENDENCIES

Dependencies in organizations kill Agility. Anytime one person has to communicate with another person it's more difficult to change. When we have small teams, managing dependencies is easier. When we have multiple teams that are working toward common goals, this inter-team communication and coordination can dramatically slow people down. When we're dealing with technical debt and defects, tight coupling, and low cohesion…the level of coordination goes through the roof and will inevitably work against our desire to get any benefit from Agile. Therefore, our ability to remove dependencies is a key part of our longer term Transformation plan.

> **ANYTIME ONE PERSON HAS TO COMMUNICATE WITH ANOTHER PERSON IT'S MORE DIFFICULT TO CHANGE**

But, dependencies are everywhere. We cannot remove them, and we cannot control them. An Agile Transformation Change Agent has two choices. Either remove dependencies or manage dependencies. You cannot have it both ways. Removing dependencies will increase agility, but unmanaged dependencies, or dependencies that are discovered late in the process, will create unmanageable chaos. If you are overwhelmed with dependencies, or dependencies are getting in the way of delivering product in a reliable and predictable manner, your transformation plan will have to address how to manage dependencies now, and how to break them in the future.



WHY

WHAT



THEORY & APPROACH



REFERENCE ARCHITECTURE

HOW

WHO

WHAT

As we get started with our Agile Transformation, it is important that we understand what's important from a theoretical point of view and that we are all clear on the patterns we intend to apply getting there. As a change agent, you will find that you may need to do some things early, while you are managing and breaking dependencies, that you don't need to do later as your transformation matures. Let's get really clear on what this looks like.

THEORY & APPROACH

At this point, we have a notional idea of why we want to do an Agile Transformation, and we have a notional hypothesis around why we want to approach the Transformation the way we do. Now, it's time to explore what the work of the Transformation is going to involve and what it'll look like when we're done. Sure, your Transformation will evolve and you'll learn along the way. Just because we don't know everything, doesn't mean we don't know anything. Let's plan for what we know and learn how to become resilient when we learn new things.

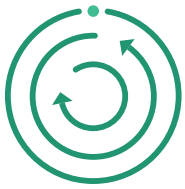
The 3 Things

Back in the early days of Agile, before the Agile Manifesto even, people were exploring better ways of working and learning better ways of building software. From those early learnings, several of the methodologies we know today came into being. Methodologies like Crystal, Scrum, Lean Software Development, Adaptive Software Development, FDD, and DSDM all existed before anyone even coined the phrase Agile. What all these methodologies had in common, what even newer scaled approaches to Agile all have in common, is that they are predicated on our ability to form teams, build backlogs, and produce a working tested increment of product at the end of some predetermined time period. The processes of each methodology were designed to enable these Three Things.

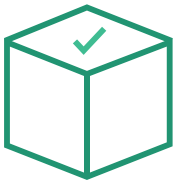
Fast forward a few years, and we have a proliferation of certifications teaching people how to do Agile. While these certifications are good and valid, and even address some of the underlying issues that might get in the way of adopting the methodology, they don't address how to systematically overcome the issues associated with forming teams, building backlogs, and producing a working, tested increment of product at the end of a time box. As a result, we have an army of Agile-trained practitioners that are going through the motions of Agile without the permission, skills, or experience necessary to make the changes necessary to really get the benefit from the certification that they recently received.



BACKLOGS
Governance



TEAMS
Structure



WORKING TESTED SOFTWARE
Metrics

The first thing to remember when designing an Agile Transformation strategy, is that if you aren't forming teams, building backlogs, and producing working, tested product at regular intervals, you aren't doing Agile. Even if you're going through the motions of Scrum.

So, let's explore what it means to form teams, build backlogs, and produce a working, tested product increment.



TEAMS

Teams are a very specific construct in Agile. They are between six and eight people, usually. That means they're small. They're independent. They're self-organizing within their boundaries and constraints. They get to decide how the work gets done and who in the team gets to do it. They are a unit that can pull a requirement off their backlog, make a commitment, and deliver against that commitment in a sustainable way. Teams are usually organized around a product or feature set, but can be organized around a component or a service, especially when large, complex enterprise solutions are involved. Regardless of scale, the fundamental rules of being a team apply.

Teams like this are difficult to form for many reasons. Sometimes it's a staffing problem in that we don't have enough of the right kinds of people to be on a single team—they have to be shared. Sometimes it's because of dependencies between teams. Sometimes requirements are shared between teams. Sometimes it's because requirements have to be coordinated across teams or multiple teams are required to deliver something that a customer cares about. Even though forming teams is difficult, it still must be done. Your Transformation plan has to start with a model or framework, a strategy if you will, for how we're going to form the right kinds of teams.

If there are impediments to forming teams, removing those impediments will be part of your Transformation strategy.

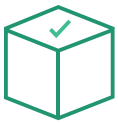


BACKLOGS

Like teams, backlogs are also a very specific construct in Agile. Backlogs are made up of stories. Stories are independent, negotiable, valuable, estimable, small, and testable (Credit to Bill Wake and Mike Cohn for teaching me this). They're small enough that a team can deliver a handful of them in a single sprint. They're not prescriptive and are open to negotiation, but clear enough that the team can estimate them and complete them within a time box. We know what done looks like, so that when we finish an item we can call it complete. They have meaning to the customer, such that the customer can sign-off on completion. They can be exchanged for one another when we learn new things.

> **BACKLOGS ARE MADE UP OF STORIES. STORIES ARE INDEPENDENT, NEGOTIABLE, VALUABLE, ESTIMABLE, SMALL, AND TESTABLE.**

Come to find out, creating backlogs like this is difficult too. The people that get to decide what goes in the backlog often aren't available to the team to define the stories or help the team get clarity. The people we do assign to the team, often don't have the authority to decide. It's common to have stories that are too big, too abstract, not definable, not able to be completed, or are too technical for the user to understand what the team is doing. Sometimes backlogs are difficult to create and prioritize because of competing demands or too much work in progress. For all the reasons backlogs are hard, they're still essential.



WORKING TESTED PRODUCT

Getting to a working, tested, validated product increment at the end of each and every time box is the Holy Grail of adopting Agile. When we can do this, the team knows that it's done. That there's no more work to do on the requirement. They know that there isn't an indeterminate pile of un-estimated work waiting for them on the backside of the project. Failure to get to an accepted done state makes it impossible to establish velocity, and if we don't have velocity against a known backlog, our project is out of control. It is impossible to measure progress without a working, tested increment of product.

Getting to a working, tested increment of product is going to be very difficult for many organizations because there are many impediments which will threaten this goal. For a small team, this might mean changing how you write requirements, how you test, and how you deploy. For large organizations, it might involve a full-scale business re-architecture, technology refactoring, and even a rethink on how you bring products to market. Again, no one said this was going to be easy, but if we're going to plan a Transformation, we have to know the fundamentals and have a plan for how to get there.

> **IT'S IMPOSSIBLE TO MEASURE PROGRESS WITHOUT A WORKING, TESTED INCREMENT OF PRODUCT**

The Three Things at Scale

When we begin to think about scaling Agile, we're fundamentally talking about scaling the Three Things. Structure is the expression of teams at scale. We want cross-functional, collaborative participation at all levels of the organization—and for all functions. Governance is the expression of the backlog at scale. It's about how we govern and manage the flow of work, decompose requirements, and make prioritization decisions and economic tradeoffs in the face of uncertainty. Metrics and tools are the expression of working, tested product at scale. It's the way that we measure how the entire organization is delivering value, not just at the team level, but across teams and across the entire organization.

If the entire organization does not succeed, it doesn't matter how well we do Agile at the team level.

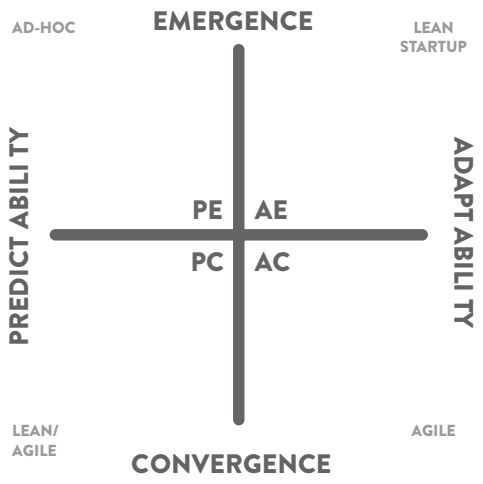
Tying Back to Business Case

Once we're clear on what we're trying to accomplish, and we have a hypothesis for how to get there… we can start to think about the fundamentals of what is actually going to change. Our goals around predictability, quality, early return on investment, cost savings, product fit, and innovation will all inform the strategies that must be defined for how we form teams, build backlogs, and produce working, tested product. The organizational impediments we uncover will inform how we plan our Transformation and give us guidance for what can be done now, and what we can do in the future as we actually improve the system by removing constraints.

The Four Quadrants

The Three Things gives us a thinking tool for understanding the fundamentals of what we're changing within the enterprise. It's all about forming teams, building backlogs, and producing working, tested product. Anything that gets in the way of this is an impediment that must be managed or removed, and all of that goes into our Transformation strategy. Sometimes though, building trust and confidence in the organization has to happen first. Sometimes we have to meet the organization where it is and show it how Agile is more effective before we can establish agency to create more—and better—change.

The Four Quadrants, and the LeadingAgile Compass metaphor, grew from a need to explain, not only why different methodologies arose to meet different organizational needs, but also how to orient the organization to where it is today and where it needs to get in the future. To understand the constraints that drive decision making around process, and how we can change these constraints over time to give us more flexibility in achieving our goals. The Four Quadrants give us a way to talk about where you are today, where you need to be in the future, and a way to layout a plan for how we'll get there over time.



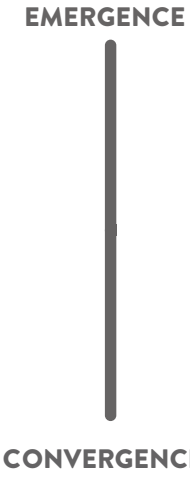
PREDICTABILITY VS ADAPTABILITY

To begin, we must recognize that organizations have competing needs. Executives need their teams to make and meet commitments. These executives made promises, and their organizations need to deliver. That said, these same executives live in a world of uncertainty. They're responding to change all the time and need their teams to respond to that change with them. The thing to recognize is that the need to be predictable competes with the need to respond to change. If we optimize for predictability, we make it harder to change. If we optimize for change, we make it harder to make and meet commitments. At this point there's no value judgement one way or the other, but recognizing where you are, and what you value, is critical to your Transformation journey.

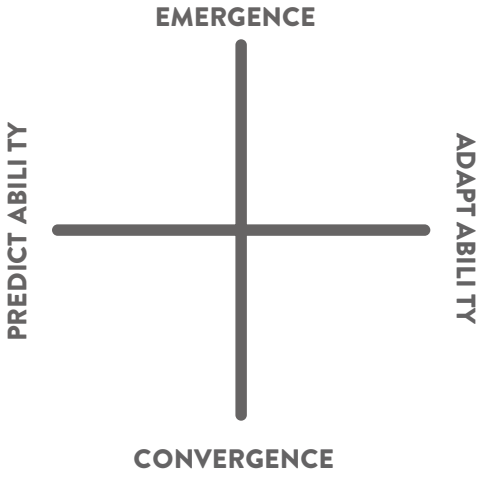


EMERGENT VS CONVERGENT

Markets have similar dynamics to companies. Some markets are undefined and tapping into those markets has a high degree of uncertainty. Sometimes the market doesn't exist, and sometimes it's so new that the requirements are'nt defined—or even definable. We call these markets emergent. Companies playing in these markets are testing product hypotheses to validate that what they think will work has a chance of actually working. Other markets are what we call convergent. They know what they want and they want it fast, cheap, high quality, and on schedule. Markets like these are not optimizing for experimentation, they're optimizing for reducing risk.

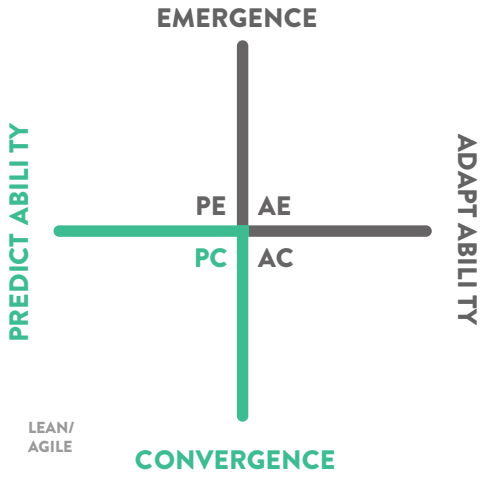


Often, the challenge on the emergent/convergent axis is that the markets with convergent expectations behave in an emergent manner. This happens because while they understand at a high level what they want and what they need, they don't have specific understanding of how to build it or what it will take to build it. Sometimes the gap is so big that it's practically impossible to put together a plan with any degree of confidence. This behavior often drives us to attempt Agile in an organization exhibiting adaptive-emergent behavior, but that values predictive-convergent outcomes.



PREDICTIVE-CONVERGENT

As the Four Quadrants model was emerging, we identified the predictive-convergent quadrant first as the quadrant of traditional project management. Over time, what we've learned is that the predictive-convergent quadrant is also the quadrant of a highly structured, governed, and predictable Agile ecosystem. It's characterized by small teams, coordinated by teams which define requirements and integrate work, and subordinate to a team that can balance investment decisions and negotiate economic tradeoffs. This is the quadrant of the more heavyweight methodologies, and the one best suited to Gartner Mode One implementations.



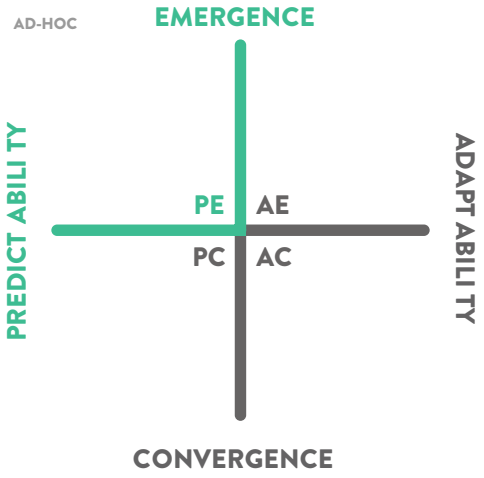
ADAPTIVE-EMERGENT

Next, we identified the adaptive-emergent quadrant. This is the quadrant of experimentation and learning. It's the quadrant of exploring and creating new markets. It's the quadrant of Lean Startup. The quadrant of innovation. It's the quadrant of Gartner Mode Two. This quadrant is characterized by small, independent teams, with very few—if any—dependencies between them. Teams in this quadrant are funded to solve problems, not to deliver specific requirements within time, cost, and scope constraints.

These are the kinds of teams that can go super fast, but you don't necessarily know where they're going to land. What you're looking for here is disruption and new markets.

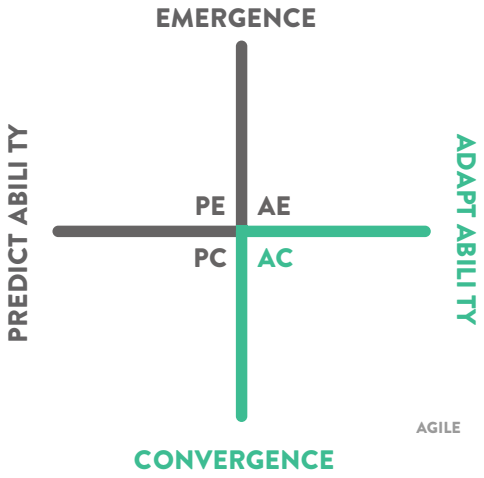
PREDICTIVE-EMERGENT

The most interesting quadrant, and the one we weren't sure actually existed when the model was first constructed is the predictive-emergent quadrant. Ironically, this is where most of the market is right now. Built for predictability, but struggling with unrealistic expectations and constant change. This is the quadrant of chaos and heroics. Companies in this quadrant deliver, but they don't often deliver what they planned. They go on death marches and depend on a handful of key people to make it happen when it really counts. Many organizations in this quadrant see Agile as the way out, but don't realize the constraints that will get in their way.



ADAPTIVE-CONVERGENT

This quadrant is Agile's home base. Adaptive-convergent is about making and meeting commitments, but in smaller batches. It's the quadrant of weekly, monthly, or quarterly delivery. It's convergent in that we value delivering scope within time and cost constraints, but it's emergent in that we deliver in such a way that we create optionality. We break projects and requirements into smaller chunks and deliver those chunks into market more frequently. We create optionality in terms of how we organize our work. We create options for how we deliver. We create options so we can change direction when we learn new things.



Applying the Four Quadrants

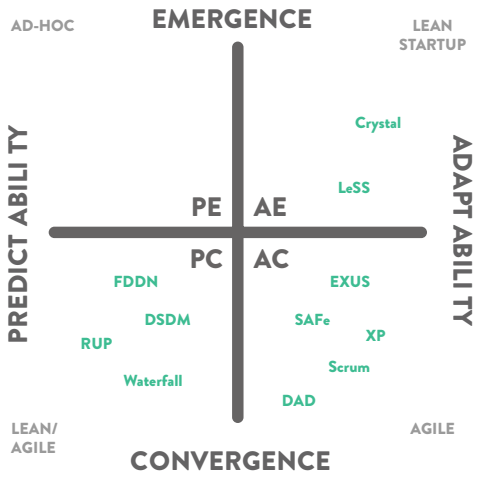
The Four Quadrants are an effective tool for getting consensus around where an organization, a division, a product line, or a team is today and comparing that to where it needs to be in the future. It provides context not only for understanding where you are and where you want to go, but for discussing the organizational impediments that are going to make the journey difficult. Often, you're stuck in one quadrant because you don't have the trust, you haven't moved the people, or you have overwhelming technical constraints that keep you locked in place.

Understanding this will help you define a process you can do today, a process you'd like to adopt in the future, and a plan for creating the conditions necessary to make those changes a reality.

Rationalizing the Methodology Wars

Every methodology ever created worked for someone. The challenge is to understand why it was created and in what context it was intended to be applied. Today, one of the biggest challenges we have in the industry is the blind application of process without regard to the organizational context in which it was designed to work. As a Transformational change agent, you can use the Four Quadrants model to identify your methodology of choice, decide if it aligns with your company and your market, determine the likelihood you can adapt your ecosystem to work in alignment with the prescribed processes, and if it will solve your problems.

Sometimes you find that your methodology choice isn't appropriate to the context, or that the organization isn't willing to make the necessary investments in the organization, process, or technology that's required to get there. Choosing a methodology and hoping for the best isn't a good Transformation strategy. It is better to align your choice with your goals and constraints and to put together a plan for getting you and your teams closer to an idealized end state.



Gartner Bi-Modal

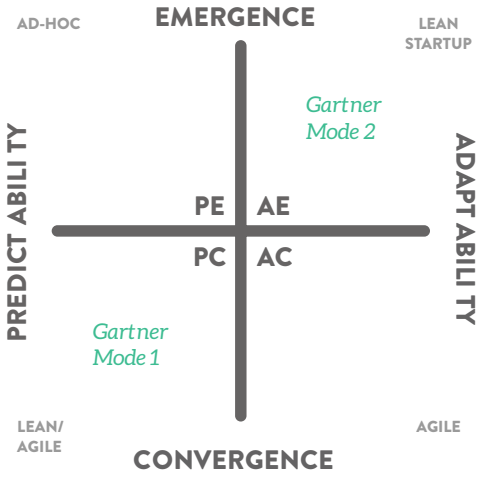
The Gartner Bi-Modal model is a popular thinking tool for executives struggling to balance the need for some parts their organization to move fast, while other parts of the organization are constrained to move more slowly and with more intentionality. Regardless of whether it's business or technology constraints, these slower, more intentional parts of the organization are unable to move at the speed of the market.

Unfortunately, this often gets interpreted as some parts of the organization will remain Waterfall and plan-driven while others will go fast and use Agile. This is an unfortunate misunderstanding. It's all due to the fact that Agile is often applied, without structure, discipline, and planning, so it's assumed that key systems and systems with dependencies won't be able to take advantage of a more Agile approach.

The key is to realize that every system can benefit from some form of iterative and incremental delivery that amplifies testing, feedback, and creates an option for low-cost change. Systems of record live in the land of Gartner Mode One and are well suited for Agile approaches that base in the predictive-convergent quadrant. These are your banking platforms and other legacy systems the can never be wrong. Systems of innovation live in Gartner Mode Two and are well suited for Agile approaches that base in the adaptive-convergent and adaptive-emergent quadrants. These are your websites and mobile applications that use the APIs from your systems of record.

> **THE KEY IS TO REALIZE THAT EVERY SYSTEM CAN BENEFIT FROM SOME FORM OF ITERATIVE AND INCREMENTAL DELIVERY THAT AMPLIFIES TESTING, FEEDBACK, AND CREATES AN OPTION FOR LOW-COST CHANGE.**

It's all about doing what works in your particular context.



Basecamps

Extending the Compass metaphor further, LeadingAgile introduced the concept of a Basecamp. A Basecamp is simply an intermediate state along your Transformation journey. Direction is set and you're guided by the Compass, while the Basecamp is an intermediate step along the way that allows you to measure progress, claim an intermediate victory, and possibly rest and refuel for the next leg of your Transformation. Remember, the work of a Transformation is changing your organizational context so that the change will stick. This can be a long process. The organization needs to move in smaller, more quantifiable steps.

Basecamps allow you to break your Transformation into smaller funding increments with measurable and defined intermediate goals.

B1 BASECAMP ONE - STABILIZE THE SYSTEM

In the presence of extreme dependencies, it is often very difficult, if not impossible to achieve true Agility. At Basecamp One you're typically looking at interactive an incremental team-based development. You can likely use Scrum or Kanban at the delivery level, but teams are often heavily orchestrated by a program or integration team. They're operating under a heavier governance model. Ideally these governance bodies are using a Lean/Kanban-based program and portfolio system—that's the goal—but roadmapping, rolling wave planning, and progressive elaboration should be the rule. Teams lose some decision making at this Basecamp for the sake of receiving highly coordinated backlogs that are aware of system constraints before the teams take stories into planning.

B2 BASECAMP TWO - REDUCE BATCH SIZE

Even in the presence of extreme dependences, there is often much work that can be done to lean out the release management process and begin improving technical practices. Once we have the system of delivery built, delivering, and stabilized…the next step is to do the work necessary to get the product into market faster. The hypothesis is that once we've improved the organization's ability to make and meet commitments, we'll earn trust with the product organization and begin to break the portfolio investment increments into smaller batches so they flow through the system faster, release to market more frequently, and give us the opportunity for earlier feedback from our marketplace. Success at Basecamp One, provides the agency to move toward Basecamp Two.

B3 BASECAMP THREE - DECOUPLE DEPENDENCIES

Many organization will find a nice, stable home base at Basecamp Two. This is the world of Gartner Mode One, systems of record, things that can't be wrong or are too expensive to refactor. Basecamp Three demands that we start breaking things into pieces, that we begin to decouple dependencies between value streams and teams. Decoupling can happen through our product and investment decisions, and how we staff teams. Often decoupling happens by refactoring our technology into services and components that can be supported by complete cross-functional teams. Business architecture drives these decisions and informs our strategy. This is often where it makes the most sense to introduce DevOps, Continuous Integration, and Continuous Deployment.

B4 BASECAMP FOUR - LOCALIZE INVESTMENT DECISIONS

Once you have completely decoupled your value streams and teams, you have the option to make changes around how you fund projects and teams. One of the biggest impediments to Agility is using the project as a funding construct when you have a product-based organization. Especially in large, complex systems of systems, the funding token spans multiple teams, multiple organizations, and multiple spans of control. Once you've done the work to decouple the organizations from each other from a staffing perspective, a business perspective, and a technology perspective, you can entertain implementing a platform/product funding strategy where shared components have their own investment dollars to support the ongoing work of the product organization. This is where you can really begin to move fast and with Agility.

B5 BASECAMP FIVE - INVEST TO LEARN

Once you have complete cross-functional teams, that are free of dependencies, and are locally funded, you can begin to change what you expect from these teams. A team that is tightly connected with the operations of the larger organization is beholden to requirements and schedules because the rest of the organization depends on them for critical infrastructure. A team decoupled form the organization across all dimensions can begin to shift how it receives and delivers requirements. Rather than a specific list of features and functions, they can begin to operate off a list of business goals and objectives, objectives that they are free to explore and test without concern for how the broader organization may—or may not—consume their services.

A Note on Basecamps

Basecamps are dynamic and should be tailored to your unique journey, the complexity of your environment, and the nature of the changes you're looking to implement. We find organizations trekking from Basecamp Zero to Basecamp Two. From Basecamp Three to Basecamp Five. Or even from Basecamp Five to Basecamp Two as their organizations mature and evolve and as the expectations of their clients change. These five Basecamps are offered as a starting place and the sequence is offered as an example of how one might proceed. Feel free to leverage and apply this metaphor in what ever way works for your organization.

> **BASECAMPS ARE DYNAMIC AND SHOULD BE TAILORED TO YOUR UNIQUE JOURNEY, THE COMPLEXITY OF YOUR ENVIRONMENT, AND THE NATURE OF THE CHANGES YOU'RE LOOKING TO IMPLEMENT**

It's worth explicitly noting at this point, that there is also nothing explicit or implied within the Basecamp standard definitions, or examples when technical practices or product management practices must be introduced. Typically, we focus on technical practices in Basecamps Two and Three and product practices in Basecamps Two and Four. But, depending on the particulars of the organization, its constraints, starting Basecamp, and destination Basecamp, it may be advisable to start introducing these concepts earlier in the Trek. This is especially true if you know you're going to pass through Basecamp Three and need time to educate the development team of legacy refactoring techniques that will take time to learn and absorb.

Remember, this is only a pattern, and metaphor…a thinking tool for change.

Expeditions

Expeditions are groupings of teams that will make the Transformation journey together. They consist of all the pieces of the organization necessary to fully implement all the pieces of the model. An Expedition should also have all the structural elements necessary to deliver the product, coordinate and overcome dependencies, as well as make prioritization decisions and economic tradeoffs. The structure has to operate in a defined governance model to coordinate and manage the flow of value at each tier of the enterprise and across the entire value stream. Lastly, an Expedition will have a metrics and tooling approach that allows the enterprise to measure, control, and truly evaluate if the Transformation is yielding the business results it promised.

Expeditions moving through Basecamps are the primary unit of progress of an Agile Transformation. Groups of teams, operating in a reliable and predictable manner, quantifiably demonstrating progress against defined business goals.

This is the goal of the Transformation effort.



REFERENCE ARCHITECTURE

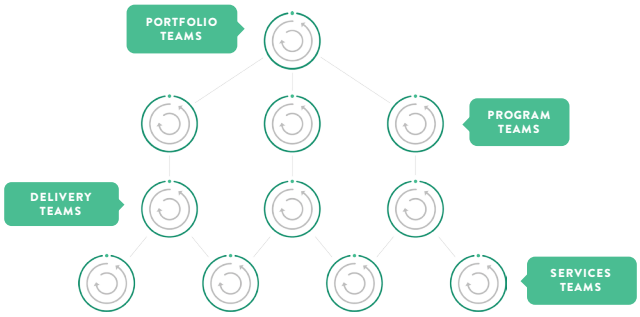
A reference architecture includes the base patterns and tools that we may want to apply within our organization as we adopt agile. This is where we start to explore what the organization might look like when we are done. It also helps us understand what the intermediate states might look like as we move toward each of the Basecamps. The LeadingAgile reference architecture includes minimal guidance around how to create structural model for your organization, a governance model, as well as model for the kinds of metrics and tools you may want to use to demonstrate value to your enterprise.

> **A REFERENCE ARCHITECTURE INCLUDES THE BASE PATTERNS AND TOOLS THAT WE MAY WANT TO APPLY WITHIN OUR ORGANIZATION AS WE ADOPT AGILE**

Structure

Your organizational structure forms the backbone for how your Agile enterprise will operate. Your structure is informed by your business architecture, your technology architecture, and your organizational chart. We're looking for opportunities to encapsulate, to decouple, and to minimize orchestration costs. Often, this mean organizing around business capabilities, value streams, and other major groupings within your enterprise. You may find that early on

in your Transformation, you can't get everything exactly they way you want it, and have to incur more orchestration costs than you'd like sometime in the future. But, this is the work of the Agile Transformation.



PRODUCT TEAMS

This is the lowest level of execution within the organization. These teams are usually six to eight people and are formed around an individual product or business capability area. In a large product, they may be formed around a subset of functionality or a feature area. A product team receives backlog from a Product Owner of a program team operating at the program tier of the governance model. The idea is that these teams should have everyone and everything necessary to deliver a working tested increment within whatever area they have responsibility to deliver.

A product team is a typical Scrum, or Kanban, team consisting of developers, testers, analysts, and other specialists that may be required to deliver against their product backlog.

SERVICES TEAMS

Services teams are a special class of product teams, and are typically responsible for an area of the product that is shared by multiple product teams. In a sense, the service that these teams support is a product in its own right, a product that is consumed by one or more other products across the enterprise. The customers of the services team are these other products and it is the job of the Services Team Manager or Product Owner to rationalize the demand across all the product areas and create services that best serve their market, as they understand it.

A services team is staffed similarly to a product team and consists of developers, testers, analysts, and other specialist that may be required to deliver against their product backlog. This team is also likely to use Scrum, or Kanban, as their process model.

PROGRAM TEAMS

A program team is a special construct in larger, more complex organizations that is instantiated to break down larger features that'll be developed by multiple product teams and/or services teams. Their job is to understand the business need from the portfolio team, bread epics into features and features into user stories. The major focus of the program team is to understand cross-cutting concerns and dependencies. They're to resolve issues that can't be easily resolved across teams and to inject user stories into the individual teams in a way that maximizes delivery flow.

A program team is different from a product team or services team in that it includes people necessary to orchestrate decisions across teams. A program team will likely consist of Product Managers, Architects, Project or Program Managers, and someone that can represent a test and validation point of view—typically a Test Manager or Business Architect. This team almost exclusively operates in a Kanban-based flow model and may not be dedicated full-time to this work. They'll likely have other organizational responsibilities.



PORTFOLIO TEAMS

A portfolio team is a group of leaders responsible for identifying business need, approving funding, and establishing time, cost, and scope constraints. This team is cross-functional like all the other teams and meets on a regular cadence to move high-level work items through the system of delivery. This team resolves prioritization concerns, makes high level tradeoffs, and applies resources to constraints to improve flow when bottlenecks are identified.

Are There Other Teams?

These four types of teams form the basis from which many of the other teams you may need are derived. Within the delivery organization, you may find that you need test and validation teams at the program level or integration teams that are responsible for providing the glue between the work product of individual Agile teams. As we look past delivery and into the broader enterprise, we may find that we need support from teams that may or may not be Agile, such as: Legal, Marketing, Sales, or Support. As we get into the governance section of this paper, we'll explore how these teams can be coordinated with the work of the delivery organization to enable greater organizational Agility.

The Relationships Between Team

These teams are usually represented as a stack, with services teams making up the bottom layer of the stack. Product teams are just above services teams, and program teams are just above the product teams. This indicates that the program teams at the top are feeding work into the lower-level teams, and demand into the overall system. There's no hierarchy here, these are merely separate continuous flow systems that are constantly feeding each other and working in harmony. Demand flows from the top of the system to the lower levels of the system in the form of requirements. Feedback flows from the lower levels of the system to the upper levels as product is delivered, requirements are further elaborated, and constraints and problems are identified.

The Impact of Dependencies

In an ideal world, every team would operate with total autonomy and independence. The challenge is that dependencies do exist, and products are inevitably bigger than a single Agile team can build within a reasonable timeframe. In many environments, there's necessary specialization at the team level due to necessary economies of scale, separation of concerns, and specialized domain expertise. All of these can drive this services team/product team dilemma.

> **THE CHALLENGE IS THAT DEPENDENCIES DO EXIST, AND PRODUCTS ARE INEVITABLY BIGGER THAN A SINGLE AGILE TEAM CAN BUILD WITHIN A REASONABLE TIMEFRAME**

In Basecamp One, the dependencies and cross-cutting concerns are explicit, visible, and managed…these are the first order concerns of the organization. As dependencies are broken, delivery mechanisms mature, and test coverage improves we move toward continuous integration and delivery, a funding strategy that decouples services development and product development, product teams and services teams become less distinct, and the program team and portfolio team constructs can often be deprecated.

Governance

Governance is the mechanism within an organization that prioritizes work, makes economic tradeoffs, determines batch size, breaks down work, coordinates across teams, manages constraints and dependencies, and deals with feedback when plans inevitably change. Governance is often associated with Waterfall style, command and control based SDLC models. While governances can be—and often is—applied in this manner, we're using governance in a much more benevolent context. At the end of the day, governance is the process through which a team receives its backlog.

SINGLE-TEAM GOVERNANCE

To drive home the point that Governance is simply the way that teams get backlog, Governance on a single Scrum team is manifest in the Product Owner role. The cadence at which Governance happens is at the sprint boundary. The Product Owner provides prioritization and makes economic prioritization by virtue of the Product Backlog and works with the team during sprint planning to decide which user stories will make it into the Sprint. While Governance is often a dirty word in the Agile community, we can see evidence of Governance in the simple processes of Scrum.



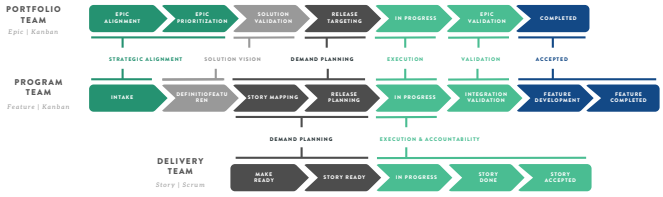
TWO-TIER GOVERNANCE

As an organization scales, there's often a need to coordinate work across teams. There are many patterns for this kind of governance. A Scrum-of-Scrums can be implemented as a proactive governance model to coordinate dependencies ahead of the team receiving user stories, or as a reactive construct where issues are discovered as they occur. The LeadingAgile Program Team construct could be used in a smaller organization, as a simple governance mechanism, explicitly instantiated with Scrum at the team level, and with Kanban at the portfolio level.



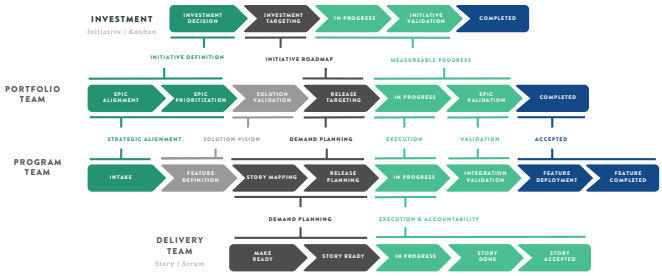
THREE-TIER GOVERNANCE

In larger environments, you'll usually see an explicit program team construct in play to decompose the work and manage the flow of value across any number of subordinate teams. A given portfolio may have many program team working together against a single prioritized queue. Each program team could have a unique Kanban board, or we could establish a common flow across all program teams. Similarly a portfolio team will operate with its own Kanban board, setting priorities and establishing flow.



FOUR-TIER GOVERNANCE

In extremely large environments, you may find multiple portfolio teams that are subordinate to an even higher level portfolio team that is managing investment strategy across many lower-level portfolio teams.



N-TIER GOVERNANCE

It's rare to find an organization that needs to maintain a queue higher than a strategy queue supporting several portfolios. At that point, you're usually above the level of a business unit, where leaders have total autonomy within the business unit, and don't have to coordinate across other business units.

Metrics & Tooling

Metrics are the way in which we measure the progress of the delivery organization. Metrics should exist at all levels of the governance model, but become particularly relevant to the business as you move higher in the stack from team-level metrics into program and portfolio-level metrics.

PORTFOLIO TEAM METRICS

- **Takt Time/Cycle Time**
  Measuring whether time to completion meets customer demand, and the duration of which the entire process—from beginning to end. This may apply in two places—service delivery time, and the development process itself, from intake to completion.

- **Time/Cost/Scope/Value**
  Cost of operations or cost of development, and the ratio of those to value received.

- **ROI/Capitalization**
  Ratio of total value received to the cost of implementation, including all planning, development, and delivery.

PROGRAM TEAM METRICS

- **Cycle Time**
  Duration from planning to completion.

- **Features Blocked**
  Number of—or total duration of—blocks in process due to impediments.

- **Rework/Defects**
  Number of times that stories are bumped back to the program team from development, and number of new issues that arise as the result of story implementation.

DELIVERY TEAM METRICS

- **Backlog Size**
  Average size of the backlog—in stories—during a sprint.

- **Velocity**
  Generally measured as total work done in an interval, such as the sum story points of all stories completed in a sprint. This should only be looked at in terms of sprint-to-sprint performance. The number itself is meaningless, but trends over time can indicate team performance as long as the story size and estimation is consistent.

- **Burndown**
  Is the rate at which work is completed during a sprint, generally measured as a trend during the sprint itself. Uneven burndown can indicate many impediments, or story sizes that are too large.

- **Escaped Defects**
  Escaped defects are bugs or problems with production software, that made it through QA – e.g. it was found by end-users or customers. Sometimes these are found in acceptance tests, and sometimes even later than that. The escaped defect rate is measured as a ratio of pre-production defects to post-production defects. A high escaped defect rate is an indication that you're moving software to production too fast.

- **Commit %**
  A comparison of the number of stories—or story points—agreed to at the beginning of a sprint to the number that was actually completed. This can be higher than 100% if development teams are under-committing, which can cause problems with prioritization.

- **Acceptance % Ratio**
  This is calculated by comparing the number of stories released to the number of stories accepted. A higher number is better in this case.

- **Scope Change**
  This refers to the number of stories where scope is changed to add or remove functionality from features in order to ensure delivery. High scope change indicates issues with backlog definition or acceptance criteria.

WHY

WHAT

HOW



CHANGE MODEL



RESULTS MANAGEMENT

WHO

HOW

Once we've established a clear sense of business goals, we have defined a working hypothesis around the nature of the change we want to make in our organization, we understand the fundamental thinking tools and the underlying reference architecture for Agile at scale. Now, it's time to begin doing the work. But, how do we actually make the changes, get the organizational momentum to move, to overcome resistance, and to make change stick?



THE CHANGE MODEL

The thing to recognize about change, is that often it's not that people don't want to change, it's that they can't see how to change. They can't see how to get from point A to point B safely and pragmatically. All they can see are the impediments, the dependencies, the resistance, the scope, and the magnitude of what has to happen to be really successful. The challenge isn't to have the right answer, the challenge is to get people to see. The Change Model is about getting people to see.

Transformation Hypothesis

PURPOSE

The reason we build a Transformation hypothesis is to get key organizational leaders aligned and rallied around a vision for what's possible.

The first set of goals involve getting agreement around the key business drivers and a shared understanding of the types of impediments that are likely to get in the way of the Transformation. The next set of goals involves agreeing on how the team will approach the Transformation, what kinds of things need to change, and how the change can happen safely and incrementally. The third set of goals revolve around looking at the where the organization exists today and determining a possible operating model within the LeadingAgile reference architecture. It is important that the leaders see their organization within the reference architecture. Finally, we need a detailed 90-day plan for what the first 90 days of discovery will look like, and a point of view for where we might pilot to get the most value as fast as possible.

> **THE CHALLENGE ISN'T TO HAVE THE RIGHT ANSWER, THE CHALLENGE IS TO GET PEOPLE TO SEE. THE CHANGE MODEL IS ABOUT GETTING PEOPLE TO SEE**

APPROACH

To build the Transformation hypothesis, we pull 15 to 20 leaders from around the company into a room. We want supporters, detractors, skeptics, people that can contribute to the organizational design, and who will be key for supporting the plan.

MECHANICS

The meeting is run over the course of two or three days and everyone needs to be in the room the entire time, We begin systematically walking through much of the material covered in the Why and What sections of this paper. The session in generally broken into eight sections, with a defined output at the end of each module:

1. **Business Drivers**

2. **Impediments**
3. **Current State**
4. **Future Structure**
5. **Future Governance**
6. **Future Metrics**
7. **Pilot Identification**
8. **90-Day Plan**

PARTICIPANTS

The specifics of who participates is often unique to the size of each organization and the span of control of the Transformation. In a smaller organization, say less than 1,000 people going through the change, you may see the CIO, COO, CFO, and key executives from Product Delivery, Human Resources, the Project Management Office, and maybe even sales and marketing.

The larger the organization, or the smaller the span of control, the lower the level of people typically are in the room. You may have a Senior Vice President, and her team, in addition to representatives from the various supporting organizations including Human Resources and the PMO. You may see leaders of specialized functions such as Quality Assurance, Business Analysis, and Architecture.

A mega Transformation, say over 10,000 people, is when you'll likely run multiple Transformation hypothesis meetings. One will be held with the most senior leadership to get alignment and support for the breadth of the Transformation, while subsequent workshops will be held with divisional leaders, and their teams, as they get closer to getting started.

OUTCOMES

The general output of the workshop is a detailed set of discovery notes which chronicle the conversations that happened over the two days. There is also a summary of findings, a summary of recommendations, a summary of next steps, and a high-level business case to support the change. The most critical outcome is that the people who attend believe the changes are possible and are willing to take the next critical steps.

Define The End State

The advantage of having a small group of senior people in the room is that they can talk candidly amongst themselves and reach agreement very quickly. The downside is that you don't get broad-based consensus and support for moving forward. The reason why the first step in the change model is called a Transformation hypothesis is that is is just that, a hypothesis, and hypotheses need to be validated. The Define the End State—or Discovery Step—is to validate the hypotheses created in the initial step, and to engage the broader organization.

APPROACH

The strategy here is to engage the target organization in one on one and group meetings over the course of several weeks, and possibly even several months depending on the size of the organization. The meetings are a way to systematically gather information, synthesize that information into progressively elaborated deliverables and plans, present feedback to the organization at regular intervals, and to slowly begin building support, energy, excitement, and commitment to proceed. The approach is based on the idea that the people you want to change may or may not know you. They may or may not agree with you. They may or may not believe that what you want to achieve is even possible. This is a relationship building step, and by virtue of progressively elaborating the deliverables and seeking feedback you build mindshare and trust with the Transformation organization.

MECHANICS

The mechanics of the Define the End State step are rather simple. With the Transformation hypothesis as an input, we begin engaging the people in the target organization in one on one meetings and group sessions, as appropriate. There should be a regular cadence with the leadership team where intermediate deliverables are formally reviewed, assumptions challenged, and risks identified. By the time the final report is created, there should be no ambiguity and no surprises.

PARTICIPANTS

Everyone in the target organization should play a part in the Define the End State step. Everyone should attend at least a single one-on-one, or group session. Everyone should be heard and have the opportunity to give feedback.

OUTCOMES

The tangible outcomes of the Define the End State step are:

1. **Product Roadmap**
2. **Organization Chart**
3. **Architectural Model**
4. **Capability Model**
5. **Stakeholder Analysis**
6. **Dependency Map**
7. **Delivery Team Model**
8. **Governance Model**
9. **Names in Boxes**
10. **Metric Standards**
11. **Tools Strategy**
12. **Transformation Roadmap**
13. **Business Case**
14. **Change Management Plan**

Intangible outcomes of the Define the End State step are increased trust, greater alignment, broad-based understanding of the necessary changes that need to be implemented, alignment around approach and timelines, as well as revised understanding of issues such as: risk, dependencies, constraints, and the business case. Ultimately, the goal is to receive financial approval to move forward. `

Pilot

Now that we have everyone in the organization engaged and participating in the change, it's time to start getting to work. The Pilot is a special case of the Expedition One to Basecamp One pattern. It involves getting a subset of the organization—a single Expedition—to a stable, reliable, and predictable state. They need to be a beacon to the rest of the organization for how the Transformation may proceed. They need to be our first reference implementation. In addition to being the first to go, much of the Transformation planning will be finalized as it is exercised on conjunction with actually doing the work. Everything we need to do for the rest of the organization will be immortalized in a document called the Transformation Playbook.

APPROACH

Assuming the target organization was fully engaged during the Define the End State change step, everyone should be clear and ready to go. The Pilot involves executing the steps in the plan. This generally involves everything necessary to get the target organization up and running and performant to the expectations of the organization. As the pilot gets underway, risks are managed, issues are escalated, and weekly progress reports are produced. Leadership is engaged for the duration of the Pilot and is actively involved in helping

to remove impediments. Learnings from the Pilot are captured and built into the structure and content of the emerging Transformation Playbook.

MECHANICS

The mechanics of the Transformation involve the mundane execution of the plan. Forming and possibly colocating teams. Getting the teams into training sessions and workshops. Getting tooling setup and configured, in addition to helping the teams get started using the tools. Backlogs are built and estimated, rolling wave planning is established, progress is assessed, and the pilot teams are hardened and stabilized over a period of several months as they move to Basecamp One.

PARTICIPANTS

Everyone in the target organization will play a part in the pilot. Everyone should be placed on a team, be a part of training, be included in the workshops, and have a clear understanding of where they fit in the new organization.

OUTCOMES

The tangible outcomes of the Pilot step are:

1. **Form Delivery and Systems Teams**
2. **Implement ALM Tool**
3. **Develop Delivery Systems Playbook**
4. **Metrics Baseline and Backlog Clarity**
5. **Develop Roadmap for 9-12 months**
6. **Make & Meet Commitments**
7. **Generate a rolling 90-Day Backlog**
8. **Improve the System of Delivery**
9. **Stable Velocity & Delivery Throughput**
10. **Build First Release Plan**

Intangible outcomes of the Pilot are similar to those of the Define the End State step. Again, the intangible outcomes are increased trust, greater alignment, broad-based understanding of the necessary changes that need to be implemented, alignment around approach and timelines, as well as a revised understanding of issues such as: risk, dependencies, constraints, and the business case. At the end of the Pilot we'll need to receive recommitted financial approval to move forward.

The Pilot will serve as a reference implementation that everyone can point to as a working example of what this should look like at scale.

ADDITIONAL NOTES ON THE PILOT PHASE

The Pilot phase, articulated here, assumes you're piloting a first Expedition to Basecamp One. That does not have to be the case. You could Pilot to a different Basecamp and Trek in any direction through the Quadrants. If this is the case, the sequence of events may change, the tangible deliverables and outcomes may change, but the intangible outcomes will remain largely the same.

General Rollout

Now that we have a working Pilot, and a revised Transformation Playbook, it's time to engage the broader organization in change. The rollout involves moving all Expeditions to Basecamp One, and likely beyond. Each new organization you engage will begin with forming a Transformation hypothesis, followed by a Define the End State step, followed by the Expedition moving to its target Basecamp. The organization is effectively looping through the Change Model steps, progressively elaborating the plan as it goes deeper and deeper into the enterprise.

APPROACH

Assuming the target organization was fully engaged during the Define the End State change step, everyone should be clear and ready to go. The rollout involves executing the steps in the plan. This generally involves everything necessary to get the target organization up and running and performant to the expectations of the organization. As the rollout gets underway, risks are managed, issues are escalated, and weekly progress reports are produced. Leadership is engaged for the duration of the rollout and is actively involved in helping to remove impediments. Learnings from the rollout are captured and built into the structure and content of the emerging Playbook.

MECHANICS

The mechanics of the Transformation involve the mundane execution of the plan. Forming and possibly colocating teams. Getting the teams into training sessions and workshops. Getting tooling setup and configured, in addition to helping the teams get started using the tools. Backlogs are built and estimated, rolling wave planning is established, progress is assessed, and the pilot teams are hardened and stabilized over a period of several months as they move to Basecamp One.

PARTICIPANTS

Everyone in the target organization will play a part in the rollout. Everyone should be placed on a team, be part of training, be included in the workshops, and have a clear understanding of where they fit in the new organization.

OUTCOMES

The tangible outcomes of the rollout step are:

1. **Form Delivery and Systems Teams**
2. **Implement ALM Tool**
3. **Develop Delivery Systems Playbook**
4. **Metrics Baseline and Backlog Clarity**
5. **Develop Roadmap for 9-12 months**
6. **Make & Meet Commitments**
7. **Generate a rolling 90-Day Backlog**
8. **Improve the System of Delivery**
9. **Stable Velocity & Delivery Throughput**
10. **Build First Release Plan**

Intangible outcomes of the rollout are similar to those of the Pilot. Again, the intangible outcomes are increased trust, greater alignment, broad-based understanding of the necessary changes that need to be implemented, alignment around approach and timelines, as well as a revised understanding of issues such as: risk, dependencies, constraints, and the business case. At the end of the Pilot we'll need to receive recommitted financial approval to move forward.

ADDITIONAL NOTES ON THE ROLLOUT PHASE

The Rollout phase articulated here assumes you are taking an Expedition to Basecamp One. That does not have to be the case. You could be rolling out to a different Basecamp and Trek in any direction through the Quadrants. If this is the case, the sequence of events may change, the tangible deliverables and outcomes may change, but the intangible outcomes will remain largely the same.

The PlayBook

The Agile Playbook is a common location for documentation, job aids, agendas and templates that support the Agile system of delivery. This includes

Agile roles and responsibilities, governance model details, metrics instructions and guidance, training material, and other artifacts.

The Agile Playbook supports Transformation by creating a common source of information and promotes sustainability over time. Such a minimal framework provides a starting point for evolving what will work best for the broader Transforming organization. The Agile Playbook is "grown organically" and as such, the initial effort is focused on creating the content essential to achieving the goals of the in-scope Transformation. The Agile Playbook is the responsibility Transformation Leadership Team.

The Field Guide

The Field Guide is a document that can be delivered in conjunction with the Agile Playbook or as a standalone document. The Field Guide is an enterprise wide working agreement for how the Transformation is expected to be managed. It outlines the expectations of the Transformation team as well as the organization being Transformed. Common Agile Playbook sections include:

1. **Who's involved?**
2. **How do we assess progress?**
3. **How does everyone stay informed?**
4. **How do we plan the Transformation?**
5. **How do we manage our Transformation**
6. **What are they key planning events and cadences?**
7. **How will we do change management?**
8. **How to escalate issues and risk?**
9. **How to escalate exceptions?**



RESULTS MANAGEMENT

Results of the Transformation are measured on two dimensions. The first dimension is the Transformation dimension. This involves measuring how the organization is progressing toward their Transformation goals. Are they learning the things that need to be learned, doing the things that need to be done, and achieving the progress they need to achieve. The second dimension of the Transformation are actual business results. Are we able to deliver product into market more predictably? Are we able to deliver with higher quality? Have we been able to reduce costs, etc…

Activities are leading indicators of outcomes. Outcomes are leading indicators of achieving a particular Basecamp. Basecamps are leading indicators of better business outcomes. In a large-scale Agile Transformation, business results can be way off. We need to create hypotheses at every level of the organization and validate how those hypotheses, those leading indicators of performance are tracking toward the end goal of the Transformational changes we're leading.

> **ACTIVITIES ARE LEADING INDICATORS OF OUTCOMES. OUTCOMES ARE LEADING INDICATORS OF ACHIEVING A PARTICULAR BASECAMP. BASECAMPS ARE LEADING INDICATORS OF BETTER BUSINESS OUTCOMES.**

Transformation Outcomes

Transformation outcomes tend to be leading indicators of lagging business outcomes. Far too often, because of dependencies, realization of business value is delayed until we can get exceptionally large parts of the organization operating in the new model.

CAPABILITY IMPROVEMENT

There are the five major categories of capability that must be present at all levels of the organization to determine if the organization is actually improving.

DEFINE THE PRODUCT - This represents the set of capabilities necessary to feed the teams requirements, decompose higher-level work items into smaller work items, and to limit batch size across the organization.

PLAN AND COORDINATE - This set of competencies represents the ability to estimate and plan work, measure velocity and throughput, assess the system for bottlenecks, coordinate across teams, and any other skills necessary to communicate, schedule, and deliver against expectations.

DELIVER THE SOLUTION - This set of competencies represents the ability to deliver a working tested increment of product to market. It could include testing practices, deployment, release management, CI/CD, and DevOps. It can be technical and non-technical

ORGANIZATION ENABLEMENT - This set of competencies includes the ability to form teams and provide the right kind of infrastructure to support and sustain those teams. This capability can be evidenced in terms of people in roles, policies, leadership, and mindset

CONTINUOUS IMPROVEMENT - This set of competencies represents the ability receive feedback and respond to change.

MEASURING PROGRESS

There are three major categories of progress that must be measured across all teams moving through the Expedition to determine if the Transformation is progressing according to plan

ACTIVITIES - Activities are the smallest unit of progress in an Agile Transformation, They're the equivalent to activities on a sprint plan. Necessary to deliver the work in the user story, but not sufficient on their own to deliver value. Activities should be tracked and can serve as a leading indicator to the progress we're making on outcomes.

OUTCOMES - Outcomes give us assurance that we're tracking to the Basecamp goals. Activities roll up into outcomes, but the outcome is the smallest unit of measurable progress on an Agile Transformation. Outcomes should reflect significant progress for a team, or set of teams, in the Basecamp. They should be measurable in the eyes of the business.

BASECAMPS - Basecamps are the highest level of value short of business outcomes for the Transformation. Once an Expedition moves to a Basecamp, its performance attributes should be extremely clear, measurable, and economically justifiable against the spend necessary to get it there.

If the Transformation is on schedule, and the capabilities that must be improved are showing sufficient progress and signs of sustainability, we'll have the right leading indicators in place to have high confidence that we'll realize our lagging indicators on the business side.

Expeditions moving to a Basecamp are the primary unit of measure in an Agile Transformation—short of measured business results.

Business Outcomes

Business Outcomes tend to be lagging indicators of behind transformation outcomes. Far too often because of dependencies, realization of business value is delayed until we can get exceptionally large parts of the organization operating in the new model. It is essential that the organization agrees that the Transformation Outcomes are likely to lead to improved Business Outcomes. If at any point in the delivery cycle, it appears this hypothesis is false, it is time to pivot and adjust the transformation.

GOALS & OBJECTIVES

PREDICTABILITY - Agile tends to focus on adaptability as a key driver, but one of the most frequent stated goals of an Agile Transformation is predictability. Predictability means that we can reliably make and meet commitments to our customers. Predictability builds trust with our internal stakeholders, our customers, and our markets.

QUALITY - As organizations scale, it's common for quality to suffer, and the way in which it suffers can come in many forms. Sometimes we're missing features and functionality. Sometimes it's extrinsic quality problems in the form of defects. Other times its intrinsic quality in the form of technical debt. Quality issues erode trust with our customers and make our software difficult to manage.

COST SAVINGS - Many companies want to adopt Agile because they believe it will be more efficient and ultimately reduce costs. Cost savings are often difficult to achieve because organizations are so overloaded with work. Often, we see organizations achieving greater assurance that their people are focused on the problems whose solutions have the highest value.

EARLY RETURN ON INVESTMENT - Many organizations are struggling with long delivery cycles that make it very difficult to get feedback from customers. It also puts a burden on the organization in the form of long investment cycles before revenue can be realized. Early ROI means that we can put product into market in smaller increments to begin charging for the product and realizing revenue.

PRODUCT FIT - One of the common goals of adopting Agile is making sure that we're building the right product for our customers. Agile gives us the opportunity to deliver in smaller batches, get frequent customer feedback, and change direction when we learn new things about our customers and their requirements.

INNOVATION - We know that well-formed teams, operating in the right market and in the right organizational context, can take advantage of Agile methodologies to exploit uncertainty. They'll be able to test product hypothesis, assess customer demand, and are free to explore what works…and what doesn't work. And, we know that truly independent teams, working in an adaptive work environment, are well positioned to exploit new markets and new ideas.

FINANCIAL MEASUREMENT

RETURN ON INVESTMENT - Measures the efficiency of your investment and looks at the net present value of your net benefits i.e. savings generated from moving from Waterfall to Agile offset by the investment made, n relation to your upfront investment. What rate of

return will you earn when making this cash outlay - the higher the return the better.

PAYBACK PERIOD - Used to evaluate when an investment will become break-even and then financially accretive. It's expressed in time, usually years or months. So, if I spend $1M how many months, or years, will it take to get my money back? After that, the rest of the money goes into the bank —the shorter the horizon, the better.

CAPITALIZATION RATE - How efficient are you in your software development process that your development team's cost can be capitalized? Do you have excess wait times or are your teams continuously spending time coding?

REVENUE ACCELERATION - As you become more Agile and get your products completed and out into the market you should see your revenue accelerate from its current growth trajectory. The revenue growth rate gets bigger/accelerates.

PRODUCTIVITY IMPROVEMENTS - As you become more Agile, your teams will increase their throughput and overall productivity. With this productivity improvement, and the ability to get more done, you can monetize it one of two ways: cost avoidance or cost reduction. Cost avoidance means not hiring employees in the future for a period of time. Cost reduction, on the other hand, means having a RIF/lay-off and gaining the benefits immediately.

EMPLOYEE TURNOVER RATE - As you become more Agile, your teams will become more engaged because they're collaborating on an ongoing basis. Higher engagement will turn into improved morale. Engaged employees are 59% more likely to stay with a company than those that are not engaged. This should reduce your employee turnover ratio.

WHY

WHAT

HOW

WHO



ROLES & RESPONSIBILITIES



SKILLS & EXPERIENCES

WHO

To achieve the outcomes necessary to run a structured and disciplined Agile Transformation effort, your company will need people to take on various roles and responsibilities within the organization. To perform these roles, the people will need to have certain skills and experiences. As you might imagine, the magnitude of change you're introducing into your organization is significant and will require more than simply Agile coaches teaching people Scrum.

ROLES & RESPONSIBILITIES

The Roles and Responsibilities necessary to lead an Agile Transformation spans executive level understanding and leadership down to people that teach team-level technical practices, and everything in between.

Transformation Lead

The Transformation Lead partners with the executive team to lead them through the Transformation process. The Transformation Lead is a high-influence peer to the executive, and as such will challenge their partner in order to produce clarity around the Transformation strategy.

They will drive the creation and execution of a Transformation strategy that aligns to the strategic goals of the organization and the enterprise Transformation strategy. In order to be successful, the Transformation Lead will need to be able to articulate the overall vision and strategy to their executive peer, to the next-level of leadership, and to the Transformation teams.

Their ultimate responsibility is to the high-level execution of the Transformation strategy and the communication of the required steps to reach the desired end state.

DESIRED TRAITS

Executive Presence:

- **Charismatic**
- **Influential and Persuasive - Seen as a Trusted Advisor**
- **Political awareness**
- **Highly Driven with Strong Follow-Through and a Focus on Owning the Outcome**

Cultural Fit:

- **High Emotional Intelligence**
- **High Integrity**
- **Strong Communication Skills**
- **High Stress Tolerance - Unflappable**
- **Self-Confidence**

Strategic Viewpoint:

- **Strong Problem-Solving and Solution Design Skills**
- **Systematic View of Organizations**
- **Driven by Continuous Learning**

CORE COMPETENCIES

Executive Level Experience:

- **Previous experience, including profit and loss responsibility, in a senior leadership role**
- **Ability to articulate where a team lies on their organization's transformation roadmap and a Focus on Owning the Outcome**

Lean/Agile Leader:

- Prior experience leading Agile transformations at an executive level
- Expertise in multiple different agile practices i.e. Lean, Scrum, Kanban, SAFe, etc...
- Capable of demonstrating organization-wide Agile adoption strategies and rollout plans; strategies for dealing with dependencies and constraints

Change Management:

- A strong point of view concerning change management, so they can articulate a high-level vision for the Transformation
- Demonstrated ability to have strong executive influence over the enterprise

RESPONSIBILITIES

- Assess the business goals and challenges to generate an end state vision for the organization
- Determine pragmatic ways to drive lasting results based on insights from the assessment
- Engage business unit leadership so they understand the Transformation, how the leadership supports the outcomes, and the impact the Transformation will have on their line of business
- Collaborate with business unit leadership to architect a situationally aware solution that achieves the end-state vision
- Works with the enablement program to enable the Transformation strategy of the organization.
- Develop plans that are outcome oriented, actionable, and reflect a deep understanding of the Agile approach
- Develop benefit cases and devise methods of demonstrating progress toward the desired benefits
- Interact with the leadership of Transformation teams to maintain alignment with the outcomes and plans
- Develop Expedition Leads over time to become future Transformation Leads



Expedition Lead

The Transformation Lead partners with the executive team to lead them through the Transformation process. The Transformation Lead is a high-influence peer to the executive, and as such will challenge their partner in order to produce clarity around the Transformation strategy.

They will drive the creation and execution of a Transformation strategy that aligns to the strategic goals of the organization and the enterprise Transformation strategy. In order to be successful, the Transformation Lead will need to be able to articulate the overall vision and strategy to their executive peer, to the next-level of leadership, and to the Transformation teams.

Their ultimate responsibility is to the high-level execution of the Transformation strategy and the communication of the required steps to reach the desired end state.

DESIRED TRAITS

Leadership Presence:

- Situational and Political Awareness
- Strong Executive Presence

- Influential and Persuasive - Seen as a Trusted Advisor
- Naturally Charismatic and People Oriented

Cultural Fit:

- High Emotional Intelligence
- High Integrity
- Team Building Skills
- Adaptability
- Persuasiveness
- Self-Confidence

Systematic Viewpoint:

- Ability to focus on the most important outcome at the current moment
- Strong problem-solving and solution design skills
- Driven by continuous learning
- Systematic view of organizations

CORE COMPETENCIES

Leadership Experience:

- Ability to build trust, empathy, and relationships with clients in any environment
- Ability to articulate where an Expedition lies on the business unit's strategic roadmap
- Ability to build trust, empathy, and relationships with clients in any environment
- Ability to sustain the Transformation team at the level required by the enterprise
- Ability to create clarity in an ever-changing environment

Lean/Agile Leadership Practitioner:

- Demonstrate organization-wide Agile adoption strategies and rollout plans including strategies for dealing with dependencies and resistance
- Expertise in a breadth of Agile practices i.e. Lean, Scrum, Kanban, SAFe, etc...

Strategic Planning:

- Ability to maintain a view of both technical needs of the team, from a software development standpoint, as well as the higher- level, strategic vision
- Ability to create continuous buy-in on the Expedition plan
- Demonstrate progress and control of the Transformation
- Ability to articulate where a team lies on their strategic roadmap

RESPONSIBILITIES

- Collaborate with the Expedition and Transformation team to develop and maintain a Transformation backlog
- Facilitate Transformation team cadence meetings
- Utilize metrics to be able to continuously show delivery progress to business
- Utilize metrics to be able to identify areas to improve delivery performance throughout the Transformation process
- Work with technical coaches & Agile process coaches to develop a cohesive team
- Work with technical coaches & Agile process coaches to develop a rolling detailed Transformation plan

- Coach teams and management towards improving Agile processes and metrics



Program/Portfolio Coach

The Program/Portfolio Coach is a critical component for groups of large, scaled product teams. The Program/Portfolio Coach is a member of the portfolio or program team who focuses on the overarching framework for the organization in question. They must be able to understand how the different pieces of team architecture, functionality, and development fits into the global structure. This requires them to be able to switch between a micro and macro viewpoint of the development of the organization.

DESIRED TRAITS

Leadership Presence:

- High Integrity
- High Emotional Intelligence
- Strong Communication Skills
- Cross-Functional Across Teams
- Ability to Maintain an Autonomous Viewpoint
- Adaptable to Quickly Changing Demands
- Strong Prioritization Skills
- Highly Driven
- Focus on Owning the Outcomes
- Strong follow-through

CORE COMPETENCIES

Experience:

- Strong experience working with Lean, must have a breadth of experience in a variety of other Agile practices i.e. Scrum, Kanban, SAFe, etc...
- Agile portfolio management experience
- Knowledgeable of new/emerging practices and coaching techniques (attendance and participation at conferences, user groups, speaking engagements, blogging etc.)
- Prior experience mentoring teams to adopt Agile practices as an external coach, ScrumMaster or Product Owner
- Project management experience, as well as a background in technical implementation to support development and architectural requirements
- Strong, flexible communications skills utilizing different mediums

RESPONSIBILITIES

- Form and coach the program team to ensure cohesive flow and dependency management
- Understand and communicate architectural needs and requirements across teams
- Maintain the integrity of the end state vision across a diverse group of teams
- Provide guidance, validation, and context to teams and assist with alignment to the end state goals
- Collaborate with the Expedition Lead to ensure that the focus for the teams remains on the end state, with the knowledge that this may constitute a shifting target
- Be able to articulate architectural and developmental specifics from teams to leadership accurately



Agile Process Coach

The Agile Process Coach educates the Expedition on Agile principles, providing them with tools and coaching for implementing the process in their own work. They conduct workshops and training sessions that teach Agile practices and ceremonies that lead toward the organization-wide adoption of Agile methodologies.

DESIRED TRAITS

Leadership Presence:

- Ability to Operate Autonomously
- Skilled at Systems Thinking
- A Strong Focus on Success and the People Required to Achieve That Success
- A Charismatic, Influential, and Persuasive Presence
- High Emotional Intelligence
- High Stress Tolerance - Unflappable
- High Integrity
- Self-confidence
- Political Awareness
- Driven by Continuous Learning

CORE COMPETENCIES

Experience:

- Breadth of experience in a variety of Agile practices i.e.Lean, Scrum, Kanban, SAFe, etc...
- Knowledgeable of new/emerging practices and coaching techniques (attendance and participation at conferences, user groups, speaking engagements, blogging etc.)
- Prior experience mentoring teams to adopt Agile practices as an external coach, ScrumMaster or Product Owner
- Highly efficient in forming a well-refined backlogs
- Ability to guide the formation of cohesive teams
- Coach teams and management towards improving Agile processes and metrics
- Ability to articulate where a team lies on their Transformation roadmap
- Relevant programming experience a plus

RESPONSIBILITIES

- Collaborate with Expedition Lead and Analyst to ensure the delivery of the Expedition plan.
- Work with Expedition Lead to form cohesive teams
- Develop and maintain a rolling detailed coaching plan in partnership with the Expedition Lead
- Coach team on Agile processes and ceremonies
- Work with teams on process improvement and metrics necessary to show progress
- Establish Agile best practices in conjunction with technical best practices advocated by the technical coaches



Technical Coach

A Technical Coach will be a seasoned software developer with a breadth of experience in multiple languages, software delivery disciplines and full systems Agile coaching. A personal emphasis on continued development of skills, as well as a familiarity with leading both from a technical and interpersonal perspective is needed, along with a passion for strong, test driven development coupled with a flexible and open view on new technical possibilities.

DESIRED TRAITS

Senior Level Experience:

- **Multi-faceted career with deep delivery, consulting and coaching experience.**
- **Experienced developer; has delivered software solutions as a senior-level architect/developer; deep knowledge of industry standards and practices.**
- **Demonstrated ability to understand client business domains.**
- **Knowledgeable of new/emerging practices and coaching techniques.**

Cultural Fit:

- **Servant leader with a Lean/Agile perspective.**
- **Pragmatic approach that can be adapted to fit business context and constraints.**
- **Ability to operate autonomously to delivery outcomes as well as to work collaboratively on a team.**
- **Operates with Integrity.**
- **Critical Thinker.**
- **Creative and Innovative**
- **Continuous career learner. Current on state of the art, and trending, approaches and technologies. Able to identify useful, new approaches and technologies as they emerge.**

Coaching & Mentoring:

- **Positive focus with a diplomatic, and constructive demeanor.**
- **Strong technical leader with a collaborative mindset.**
- **Excellent interpersonal communications skills.**
- **Understands the value of diverse teams with differing levels of competency and** the need to mentor and level up all team members. Comfortable mentoring team members at all levels of competence and experience.

CORE COMPETENCIES

Senior Level Software Architect/Developer:

- **System, solution architecture and design skills.**
- **Design and deliver solutions utilizing object-oriented and functional design paradigms, and design patterns.**
- **Polyglot developer; full stack experience.**
- **Able to drive and maintain code quality.**
- **Test-first, test-driven approach.**
- **Refactoring.**
- **Pair programming and cross-disciplinary pairing.**
- **Build and test automation**
- **Legacy code remediation**

Systems/DevOps:

- **Continuous integration**
- **Virtualization, cloud, and infrastructure as code**
- **Development pipeline and continuous delivery**
- **Competitive candidates will have experience with Pivotal Cloud Foundation (PCF)**

Lean/Agile Practitioner:

- **Works effectively with Kanban and Scrum approaches, and coaches and mentors teams to improve their Lean/Agile approach. This includes concepts and activities such as improving flow, limiting work in progress, backlog definition and refinement (e.g. story breakdown), and fast delivery of value.**

RESPONSIBILITIES

- **Collaborate with Expedition Lead and Transformation coach to understand the Expedition roadmap and necessary outcomes.**
- **Develop and deliver coaching plan to get Expedition teams to their targeted outcomes.**
- **Effectively detect, communicate, and escalate organizational, technological, and other impediments.**
- **Teach and mentor Expedition team members on Lean/Agile approaches for flowing work.**
- **Teach and mentor Expedition team members on XP engineering practices, including TDD, CI and pairing. Facilitate code reviews, provide leadership perspective on design and code quality.**
- **Coach Expeditions within the system of delivery to implement DevOps practices and technologies, including delivery pipeline design and development, cloud technologies and infrastructure as code, and continuous delivery.**



Analyst

The Analyst will support the team, providing tactical support for the overall Agile Transformation. Knowledge of Agile processes, as well as ability to execute on required deliverables is key to filling this role. The Analyst will be involved in collecting the metrics that will be used to outline the Transformation from a data-driven perspective. This person will analyze data provided by the teams to populate metrics that can inform decision making at the higher levels. This data-driven role focuses on being able to articulate status based on a well-established data pipeline.

DESIRED TRAITS

Leadership Presence:

- **Ability to Operate Autonomously**
- **High Integrity**
- **High Emotional Intelligence**
- **Highly Analytical**
- **Extremely organized with high attention to detail**
- **Highly driven with strong follow-through and a focus on owning the outcome**
- **Ability to focus on the most important objective at the current moment**
- **Driven by Continuous Learning**
- **Adaptable**

CORE COMPETENCIES

Experience:

- **Exposure to Agile methodologies**
- **Project management exposure, as well as an understanding of software delivery and technology operations processes and terminology**
- **Experience with Agile Project Management tools is a plus e.g. Rally, Jira, VersionOne, LeanKit, etc...**
- **Strong communication skills, both verbal and written**
- **Highly skilled in MS Office**
- **An understanding of creating process and other flow diagrams**
- **Business process improvement experience**

RESPONSIBILITIES

- **Synthesize and analyze data**
- **Create and maintain engagement deliverables**
- **Help design, document and maintain artifacts needed in Transformation Office**
- **Process Improvement**
- **Collaborate with Expedition Lead and Agile Process Coach to prioritize the Transformation team's Expedition plan.**
- **Responsible for collecting and distributing meeting minutes**
- **Reviewing, cataloguing, and sorting supplied data**
- **Meeting organization and planning**
- **Metrics and reporting maintenance/catalog**



SKILLS & EXPERIENCES

Now that we have the roles and responsibilities understood, it's time to consider the skills and experiences necessary to lead the change. As we mentioned, simply having knowledge of Agile is insufficient to lead the changes required for sustainable Transformation. Different levels of team members will require different types of skills and have a different strengths profile.

Team Attributes

Over the past several years, LeadingAgile has learned a ton about the skills profiles necessary, not only to support or lead a Transformation effort; but to be happy, grow, and thrive in these types of roles.

The people you have on a Transformation project need to have basic skills and experiences to do the job. That said, these are dynamic roles, and what we find is that when people are under pressure, they go back to whatever is their default mode of operation is. Because of this, understanding what people believe and how they behave is super important. It's their home base.

Leading people through change requires deep understanding of people, a high level of empathy, and an ability to create dynamic and creative solutions on the fly to help them solve their problems. For this reason, emotional and intellectual intelligence are key success factors. The ability to read people, understand their concerns, and to develop solutions in real time is critical.

The senior people in your organization must be able to, not only see the underlying patterns, but help others see them as well. In order to craft a Transformation

hypothesis, they have to be able to see the possibilities through the noise. And, they have to be able to see the system in order to help others understand the system.

SKILLS

Skills are probably the most obvious and straightforward attributes of a person to evaluate. Skills can be identified, enumerated, tested for, and interviewed for. You can see skills in action. Skills for any level of Transformation are clearly important. Generally, anyone performing a role on an Agile Transformation needs to be steeped in the Agile literature and they need to have some amount of expertise in all the major methodologies. A Transformation coach needs to be able to not only understand the topic, but be able to teach it to others.

EXPERIENCE

Experience appears to be straightforward to measure, but in the Transformation world it can be a little difficult. The first thing to consider is that doing Agile is different from installing Agile, and that is different than leading change. Even if someone has experience installing Agile and leading change, the dynamics of your organization might be different from the dynamics of the enterprise where they got their experience. We find that many leaders have a very different experience coming into a new role in a new organization. Ideally, your Transformation coaches have experiences solving a wide variety of problems and have learned to overcome a wide number of potential impediments.

CULTURE

Culture is unique to your environment and whomever is on your Transformation team will need work well within your organization. That said, there are some unique attributes required to play a role in the Transformation team. First, and foremost, a Transformation team member needs to be honest and transparent. Making sure that people understand the truth and have all the necessary information builds trust, and trust is the currency of change. Also, this team member will need to be on message, be a team player, and be collaborative. Your Transformation team, as a whole, needs to be thoughtful, and have a strong point of view, while maintaining the ability to be flexible and build consensus.

COMMUNITY

Ideally, your Transformation team members are active in their community, speaking at conferences, sharing what they know through blogs and podcasts, and are continuously learning.

BELIEFS

Agile coaches, and especially Agile coaches focused on Transformation and change, tend to fall into one of four camps in terms of what they believe about leading change. Some believe that change begins at the individual and team level, and grows within the organization. That change is about the people first and making sure they have what they need to be successful. Others believe that change can be led at the organizational level, but that it's still crowdsourced and people driven. A large majority focus on installing team level practices, and take a small scale delivery focus to leading change. It's a business focused approach, but directed in the small. Others see systemic issues and want to attack the root of the problem and are willing to take a Chief Architect type approach to solving the problem and leading others forward.

While this is an overly simplistic view, what people believe about change, and how to lead change, really matters when you're building a Transformation team. Every Transformation needs a focus… a transformation hypothesis, if you will… and the belief system of your transformation team needs to reflect and support that hypothesis

BEHAVIOR

Behavior analysis is another way of anticipating how people are going to perform when they're in the middle of a Transformation change initiative. There are some very knowledgeable people, that when their ideas are challenged will go and hide, possibly getting their feelings hurt, and taking a long time to reengage. Others are full of energy, but have low empathy and demonstrate bullying behaviors when trying to get others to do new things or adopt new ideas. LeadingAgile uses a tool called PeopleDNA and has a ton of data around what behavior profiles contribute to long term individual success.

INTELLECTUAL INTELLIGENCE

LeadingAgile uses IQ as an indicator of a Transformation team member's ability to dynamically build new solutions, based upon synthesis of past experience and their library of knowledge. We find that people with lower IQ profiles will tend to be anchored on one way that they know and will be hesitant to change when presented with new information. Higher IQ profiles are often more open to change and can process new information in real time to solve problems.

EMOTIONAL INTELLIGENCE

In a nutshell, emotional intelligence is the ability to read what another person is thinking, feeling, and experiencing. It's an ability to see the world from another persons emotional point of view. Change is scary, and people going through change are often afraid. Sometimes having the right answer isn't enough. It's communicating that answer in a way that creates safety, builds trust, and demonstrates empathy. Getting people to do new things, and behave in new ways, is seldom a purely intellectual exercise.

PATTERN RECOGNITION

Pattern recognition is the ability to see order in the chaos. It's the ability to recognize what's truly important and move forward, possibly in spite of conflicting information—or barriers—that appear to be in your way. It's being able to see the Mona Lisa on a blank canvas. It's the ability to see Venus in a block of marble. It's the ability to see the pirate ship in the Stereogram. Being able to see what is possible and lead others toward that vision is key for senior team members.

SYSTEMS THINKING

Systems thinking is the ability to see the whole, and to envision how making changes to one part of the system, creates ripples through the rest of the system. The design of large-scale organizations is definitely a systems problem. Orchestrating large-scale organizational change is a systems problem as well.

Skill Set Mix

Different people will be at different levels of skill and experience, demonstrate different beliefs and behaviors, have varying levels of IQ and EQ, and their ability to recognize patterns and systems think will be vary as well. As you being to build your team, and develop your talent, you'll find that some attributes are easier to develop than others. And, when hiring…that it's sometimes best to get the right profile mix right out the gate.

JUNIOR TEAM MEMBERS

Your junior team members are most likely to play an analyst role or an Agile practices coaching role. In nearly all cases, you want to optimize team selection for intellectual intelligence and emotional intelligence. One could argue whether to not IQ and EQ can even be developed. Regardless, developing these baseline attributes will take time. Your team members should score as high as possible in these areas, no matter what role they're playing.

Positive behavior profiles are another attribute you'd like to build your team around. People that are high energy, goal oriented, outcome focused, don't get their feelings hurt easily, and are resilient make the best change agents. You need a team of people that are able to maintain a strong point of view, but are flexible, open, and coachable. Junior team members need to have a higher respect for authority, higher attention to detail, and follow-through.

High-potential junior team members will learn what it takes to do Agile over time. They'll need mentorship from senior team members, access to books and training, and the opportunity to try out new skills on a regular basis. As they gain experience, pattern recognition becomes easier, and you'll begin to see their ability to systems think improve. Additionally, these junior team members will need the opportunity to get out and engage with the community.

SENIOR TEAM MEMBERS

Senior team members are more likely to take on more advanced and strategic roles on a Transformation project, likely playing the role of Program/Portfolio Coach, an Expedition Lead, or a Technical Coach. As mentioned in the previous section, IQ and EQ are necessary attributes of any successful Transformation coach and should be screened for in hiring, or staff selection.

The behavior profiles don't shift much either as you go from junior to senior roles, although you do begin to see a shift in emphasis. It's less important for senior people to have high respect for authority, they can tolerate having less attention to detail, and lower follow-through. The reason for this is that they are operating at a more thematic level and other people on the team are handling more of the lower level details.

Senior team members need to have an expert-level understanding of Agile, Agile methodologies, and have a broad set of tools and techniques at their disposal. Senior-level people need to be able to apply these ideas in situationally specific ways, tailoring them to the needs of their organizations. They must be knowledgeable at all levels of scale and know how to build and design systems.

EXECUTIVE TEAM MEMBERS

Executive team members will play the highest level roles and have the most influence in the overall Transformation strategy. They're playing the role of Transformation Lead, or maybe they're the sponsoring executive, or an Account Lead. These people should

be held to the highest standard of intellectual and emotional intelligence. These people are going to have to play a broad leadership role on the Transformation, solve problems at an executive level, build consensus amongst peers, and be aware of the political dynamics of the organization.

Behavior profiles at this level shift towards high influence and high resilience, while shifting away from respect for authority, attention to detail, and followthrough attributes that help the more junior people. It's important at this level that people can create and hold space. There is an art to balancing the needs of the Transformation with the needs of the organization. Behavior has to be incredibly aware, nuanced, and applied in such a way to achieve critical business outcomes.

Executive level people will benefit from deep Agile knowledge. The best ones will be able to operate as Business Architects or Technical Architects at the most senior levels. What's critical is this expert's understanding of organizational dynamics, politics, change patterns, change management, pattern recognition, and systems thinking and awareness. After all, these are the people that will be creating the space necessary for the Transformation to occur.



A FINAL NOTE ON LEADING CHANGE IN DIFFICULT ENVIRONMENTS

Most of the executives we talk to understand that the core challenges in their organizations are not solely process related. While process may exacerbate some of the dysfunction, these executives recognize that they have alignment problems. They have delivery problems. They have accountability problems. They have transparency and visibility problems.

Furthermore, solving for these problems is highly political, expensive and risky. Some people will be threatened by the changes. Some people will want to resist the changes and wait for things to go back to normal. Change is difficult under the best of conditions, let alone when you are trying to fix the underlying delivery mechanisms for your entire organization.

People will feel at risk and they will be afraid.

The problem is that if we don't address the fundamental issues truly driving the need for change, the change will be empty and ineffective. You'll burn the same amount of political capital, spend the same amount of money, take the same amount of risk, and still fail.

The LeadingAgile Transformation approach is designed to educate leaders on how to prepare for and manage an Agile transformation initiative. It is designed to create space for change to take place and help you manage your business through these changes with minimal economic disruption. Change is never easy. Our goal is to make it as safe, predictable, and as plan driven as possible so your ream will support the changes and help make them successful.

WHO IS LEADINGAGILE?



LeadingAgile is a company dedicated to helping larger, more complex organizations achieve better business outcomes through the systematic application of Agile delivery methods across the entire enterprise.

Our company is primarily focused on the transition patterns and intermediate states necessary to safely and pragmatically lean any company through a structured and planned Transformation event.



WHO IS MIKE COTTMEYER?

LeadingAgile founder, and CEO, Mike Cottmeyer is passionate about solving the challenges associated with Agile in larger, more complex enterprises. He and his team are dedicated to providing large-scale Agile Transformation services to help pragmatically, incrementally, and safely introduce Agile methods. He spends most of his time leading and growing LeadingAgile, doing sales and business development, developing content, and providing strategic coaching for key clients.

Mike is married to Kimi and has three sons Zach, Daniel, and Noah. He has an ever growing collection of guitars and an irrational obsession with Collective Soul.