

How to Scale Agile Software Development with Product Teams in the Enterprise

A Field Guide



Table of contents

Introduction	3
Stage 1: Experiment – Start small, with a focus on learning	4
Vision, mission and principles: Foundational tools for change	6
Stage 2: Adapt, Fit, Grow – Get more people working on larger, more complex problems	7
Adapt practices	7
Portfolio management & leadership	7
Stage 3: Expand – Bring change to an entire line of business.	9
Communities of practice	9
Lean governance	9
Growth boards	10
Executive sponsors	10
Stage 4: Replicate – Bring change to other lines of business	10
Seed teams	11
Assess your progress	11
Assess your business impact	12
Continuous improvement is critical for success	12
About the authors	12

“Agile transformation is not just a technology transformation, it’s a business transformation.”

RICHARD LEURIG
SENIOR VICE PRESIDENT
CORELOGIC

Introduction

Many enterprises have enjoyed amazing results by adopting agile methods. But that success is not without its challenges. Indeed, increasing code quality, shipping value faster and getting better software outcomes is relatively straightforward when you’re one small software product team. It’s much harder to do with 100 teams or more.

How do you, as a large organization undertaking an agile transformation, manage dependencies while maintaining the autonomy of each team? How do you communicate progress to a large, diverse set of stakeholders? How do you ensure security and resiliency for millions of users? How do you maintain transparency and shared ownership for a globally distributed team? And perhaps most importantly: Which outcome deserves the most focus now, and which metrics should you strive to impact?

These questions naturally arise as enterprises grow their understanding of tried-and-true modern software development practices. Here’s the good news: Answers to these questions are beginning to emerge. Agile practitioners have been continually refining the state of the art, resulting in more accessible patterns and practices. At VMware Tanzu Labs, we’ve helped many of the world’s largest and most complex enterprises apply modern software development methods at scale. They’re now building better software, delivered by thousands of productive (and happy) engineers, that delights millions of users.

How do some of our largest clients embark on large-scale transformation? By starting small and iterating, based on what they learn, along the way. Indeed, we’ve found that the same build, measure, learn approach that is at the core of how we develop software is directly applicable to agile transformation. In this paper, we’ll detail how we’ve helped organizations successfully navigate transformation. We’ll share a view of four stages of transformation, walk through the capabilities we’ve found to be essential, and provide detailed guidance on how to get started.

Stage 1: Experiment – Start small, with a focus on learning

You've decided to try a new approach to software development to get better results. Where to start? Rather than throw your full weight into large-scale change up front, we suggest starting with a small, focused and relatively quick experiment. While there is no one perfect recipe for successful large-scale transformation—and while inspiration can be drawn from a variety of places—in order to reduce waste and increase the odds of success, those championing change must proceed with an experimental mindset, as validating or invalidating hypotheses, learning and iterating are all critical.

Your initial experiment should involve one software development team (typically 8-10 people), a problem to solve, a hypothetical solution, and a metric to understand if that solution is the right one. Your hypothesis might look like this:

Our inability to quickly iterate on software and push new features to market is limiting our organization's growth. We believe one way we can solve this issue is by employing agile techniques and providing a clear path to production.

Your solution might look like this:

We'll dedicate a single team, which will employ agile techniques and be provided with a clear path to production, to developing one new application.

Then, the most important part: What evidence will you seek to help you know whether this solution is effective? Perhaps in this case it looks like:

We'll see a business win quickly and, one week later, we'll incorporate feedback and build upon that win with another push to production.

This experiment must be rigorous in order to influence change across the organization, and it should meet the following criteria:

- **Deliver a business win** – Impact a major metric that is valuable to the business, and deliver the “wow” factor that will inspire others to adopt a new way of working.
- **Demonstrate iterative delivery of value** – Show that the impact can be maintained over time through new delivery of value at a regular cadence (i.e., weekly iterations).
- **Define success at the start** – Start with desired product outcomes, identify metrics, and measure those regularly, rather than waiting for them to emerge. Software delivery performance is also worth watching. Low volatility (which measures how much a team's velocity changes week over week) and high deployment frequency indicate that a team is capable of predictably and frequently delivering new value and benefiting from feedback.

WHERE DO I FIND GREAT PRODUCT MANAGERS?

Product managers shepherd the delivery of business value by understanding the most valuable outcome to be delivered by the team at any given time, and ensuring the software being developed is aligned with that outcome. Many organizations we work with do not yet have defined product management disciplines, and wonder where they can find people who'd do well in this role. We've found that product managers frequently emerge from software development, business analysis, project or program management, and product design, but they can emerge from other places, too. Markers of a potentially great product manager are a passion for delivering outcomes, an obsession with metrics and measurement, and strong interpersonal skills to motivate teams and engage stakeholders. Read more [here](#).

PSYCHOLOGICAL SAFETY

Scholar Amy Edmondson coined the term *psychological safety* to describe the quality of relationship between employee and employer in high-performing teams, and we've found it to be absolutely essential. When teams do not feel psychologically safe, they will not take the risks you need them to take in order to learn, grow, and deliver value.

Let's go back to the solution being tested in this experiment and unpack it.

You've decided to dedicate a single team, which will employ agile techniques and be provided with a clear path to production, to developing one new application. So, which agile techniques that have been valuable for others should be included in this first experiment? What other learnings, from organizations like yours, should you incorporate into it?

In order for the team to succeed, it needs to:

Have a strong business need/problem to solve using software, one that is demonstrable and requires additional investment.

- Create a minimal viable product (MVP) in a short period of time, ideally less than three months. (What's an MVP? Here's [one way](#) to think about it.)
- Articulate the MVP within the context of an outcome-oriented roadmap that becomes part of regular conversations with stakeholders.
- With an outcome-oriented roadmap, the team will begin changing the conversation with leadership from "What features are being delivered, when?" to "What results are we on track to deliver, and what have we learned to de-risk our success?"
- Be balanced, staffed with influential people who want to try something new and evangelize to others. A typical team should include a product manager, a product designer, and anywhere from two to six software developers. See the sidebar for more on sourcing the right product manager for your first team.
- Ensure psychological safety, both currently and going forward (see sidebar).
- Design a clear path to production so that it can ship as soon as valuable features are ready.
- Empower and support a single, bold leader and provide executive support. The path to progress will not be linear; it will feel chaotic at times. A bold leader, one who can make hard decisions with imperfect information and is willing to go against the political grain and ruffle some feathers, is critical.

Run your experiment, in full, to gain learnings. Run it anywhere from one to three times, with new teams, and seed each new team with at least one person from a team that's already been through this type of experiment. You'll make several assumptions about how impactful your new way of working will be on outcomes delivered. Ultimately, you're after a business win and quick iterations. In many cases, we've seen transformation leaders be required to deliver evidence of wins/proof points in order to gain funding that will enable them to move on and expand beyond a team or two. Be sure to clearly define success, so that after it's been achieved it can't be written off as a fluke when the time comes to influence change at scale.

While focusing on your experiment, don't wait until the next phase to start thinking about the things you'll need in order to be successful as you expand. They take time. Note, however, that while these are **not** critical to have in place by the end of Phase 1, they will need to be in place by the end of Phase 2.

WANT MORE INFORMATION ON GETTING STARTED WITH AGILE? CHECK OUT THESE RESOURCES:

- [Tanzu Labs' modern application development services](#)
- [Your Path to Agile Matters](#)
- [Digital Transformation in the Wild: A Look at The Home Depot](#)
- [Scaling Leadership of an Agile Software Factory: Lessons Learned](#)
- [How Communication Makes Agile Development More Effective and Efficient](#)
- [4 Secrets to Remote Agile Success](#)

Vision, mission and principles: Foundational tools for change

Successful change requires an understanding of desired future outcomes, contrasted with the current state, and a plan for how to get from point A to B. Successful organizations know where they want to go, have a strategy for how to get there, and define ways to measure their progress.

Start by developing for your work:

- A **strong foundational understanding of the current state** of your organization, namely its people, process and technology. This will illuminate any pain points, barriers and opportunities as they relate to efforts to effect change within the organization. Value stream mapping can be especially useful here.
- A **vision** – Think about both the ultimate results that will be delivered by the team and the impact those results will have on the organization. Think as well about the culture you aspire to create, and what your larger organization might look like once it's transformed. This will be ongoing work, but for the sake of the exercise of developing a vision, it can be helpful to think about what things will look like when you're done.
- A **mission** – Your mission should capture your core value(s), be something you're ultimately striving for, and support your organization's overall vision. When you're done delivering on your mission, your vision has become a reality. Mission statements help organizations frame and prioritize work by considering whether and how closely that work supports delivery of the mission.
- **Principles** – Developing an effective and sustainable approach requires more than following predefined organizational structures and process maps, however. To drive lasting change, focus instead on strong, foundational principles that are aligned with your organization's values. As an example, we have listed the VMware Tanzu Labs principles.¹ These principles are shared by everyone within our organization, and are at the core of what we do. They provide us with a strong foundation as we strive to adapt quickly to feedback.



EMPOWER TEAMS

Diverse teams with shared context, psychological safety and decision-making authority move faster.



START SIMPLE

Learn just enough to begin and let the work inform the direction.



EMBRACE CHANGE

Expect change and turn it into opportunity.



DELIVER EARLY AND OFTEN

Ship value to users and measure impact to lower risk.



IMPROVE CONTINUOUSLY

We are always learning and humbled by what we don't know.



GIVE BACK

Share our time, tools, code and approach with internal and open source communities to maximize impact.

Let's assume you've run your initial experiment and gathered evidence to support a plan for expanding your new way of working. What's next?

1. Current at the time of publication. Principles may be revisited and adapted if and when needed.

REMOTE WORK

Inspiring great outcomes from remote teams is now an essential capability for any leader. Over the past several years—and particularly in 2020, when COVID-19 forced nearly every organization to embrace some level of remote work—we've learned a lot:

- [Need Help Working Remotely? Check Out Our Tips](#)
- [Resources for Remote Software Teams](#)

Stage 2: Adapt, Fit, Grow – Get more people working on larger, more complex problems

Once you have evidence that modern software development methods will yield better results, you're ready to expand to more teams. In this phase, you'll expand the success demonstrated by your experiment team to an additional three to six teams. However, you've likely already encountered inertia and resistance from parts of your organization. The continuous improvement of software—and the continuous improvement of the process for developing that software—takes discipline and a willingness to change. But people are naturally resistant to change, and are rarely motivated by others simply telling them that better results are possible. Rather, they need to be inspired and involved.

At the start of this phase, you'll need a **strong** vision for your transformation effort as well as alignment around the principles that will guide your organization's new culture. You started thinking about this toward the end of Phase 1, and now is the time to finish that work.

Now is also the time to identify **leaders** who *cultivate the strengths of individuals on teams*, ensure that effective practices flourish, and inspire others to change.

But first, pause to consider what adaptations need to be made to your new way of working so as to increase adoption. We cannot stress enough how important it is to iterate based on learnings. Especially since you now have two primary internal goals: 1) support and grow a new way of working that has delivered proven benefits, and 2) drive adoption throughout the surrounding organization that is not used to this new way of working (and may not have experienced a benefit from it yet). Adapting the new way of working just enough to help expand it through more of the organization is the way you'll achieve both goals.

Adapt practices

With your first experiment or two, you might have taken a pure approach to agile (e.g., with XP or scrum) that helped your team deliver better outcomes. But if the rest of your organization works differently, their adoption of a radical new approach will likely be difficult and slow. We recommend making some strategic adaptations to any pure agile approach you take in order to make widespread adoption faster and easier.

Team colocation, for example, is central to many pure forms of agile and may have worked well for your initial short experiment. But given the increasingly distributed nature of work, colocation can be challenging to handle at scale. Since colocation is used to tighten feedback loops and thereby reduce waste, consider how else you might achieve that goal. Investing in exceptional remote tooling is one way, as is ensuring that primary teams and their fellow, dependent teams are always available to one another for some number of hours each day.

Portfolio management & leadership

When you have three to six teams that must work together to deliver outcomes, you're dealing with some new challenges:

- Your teams are working together to deliver value, but they're not delivering the desired outcomes
- You're shipping great software, but your downstream teams are raising red flags about missing data, broken integrations and other negative results
- Your product teams are being slowed down by the need to manage communication dependencies, orchestrate technology and cross-cutting features
- Your product managers are overworked, and there's conflict between product teams over priorities and resources

Facilitating the work of multiple product teams working together to deliver shared product outcomes is no small feat; it requires sequencing backlogs and managing detailed product roadmaps. In our experience, new roles are needed to ensure that teams are aligned with the portfolio and outcomes for each line of business, and to make sure the teams are healthy and working in a sustainable way. Portfolio management refers to a set of practices designed to get multiple teams dependent on one another to provide both the business and the end user with substantial value. Successful portfolio management requires the dedicated focus of portfolio leaders.

Portfolio management involves setting the vision, strategy, roadmap and related metrics for a portfolio of applications. In terms of metrics, they will be a combination of DORA metrics (to measure software delivery performance) and product metrics (to measure product-market fit). Portfolio management also involves orchestrating teams to make connections and deliver portfolio-level outcomes. In an agile environment, where individual teams are constantly learning, adapting and iterating, creating a centralized source of vision and strategy for the portfolio—and portfolio-level outcomes—can be a lot of work and, in our experience, requires dedicated oversight by a portfolio leadership team.

A portfolio leadership team provides multiple benefits, by:

- Removing blockers
- Maintaining a shared understanding among teams
- Prioritizing shared or dependent work
- Preserving context across a number of products
- Identifying opportunities

How you organize portfolio leadership will depend on the size and complexity of your portfolios, but there are some common responsibilities you'll need to consider. A typical structure we've seen be successful is comprised of:

- **Portfolio product lead:**

- Works with other portfolio leaders to prioritize outcomes
- Manages stakeholder relationships and business priorities that concern all portfolio products
- Does not pass requirements down to product managers

- **Portfolio design lead:**

- Works with other portfolio leaders to prioritize outcomes
- Harmonizes the user and customer experience of the portfolio
- Does not enforce consistency for consistency's sake

- **Portfolio engineering lead:**

- Works with other portfolio leaders to prioritize outcomes
- Illuminates data outcomes, non-functional requirements and technical opportunities
- Does not require detailed architecture up front

How, then, do you connect portfolios to one another? The following additional roles come into play:

- **Directors** are responsible for strategic business operations, not the success or failure of products or portfolios.
- **Program managers** connect the dots among portfolios. They also focus on a particular discipline (i.e., software engineering, design) and are responsible for coordinating impacts and dependencies between teams as well as helping teams achieve aligned autonomy.

CUSTOMER STORIES

We've helped a number of enterprises get better results from software by applying techniques like the ones described in this paper. Here are a few stories:

- [Digital Transformation in the Wild: A Look at The Home Depot](#)
- [Developing Software As a Team Sport \(Dick's Sporting Goods\)](#)
- [Iterative Innovation Incubates at DBS](#)

You'll also need change agents to foster the growth of people and new practices, and to provide leadership to the growing number of people working in the new way. Such change agents must be skilled at managing up and out by communicating and coordinating the expectations and inputs of investors, customers and stakeholders.

Source these leaders from your current pool of practitioners. Existing leaders will have context and empathy for any external teams that might be challenging the change you're introducing. They will also have credibility with anyone they seek to bring along.

“With [Tanzu Labs] we have minimized our innovation cycles and can now respond to changing market demands faster than ever before, both of which allowed us to develop a profound software culture in our transformational journey into a digital company.”

CHRISTOPH HARTUNG
HEAD OF CONNECTED CARS
[MERCEDES-BENZ](#)

Stage 3: Expand – Bring change to an entire line of business

Now you're working toward the next outcome: a successful line of business that delivers results driven by portfolios of applications. You're delivering better business results, so you're ready to expand what's working well with 3-6 teams to dozens of teams within the line of business. You're also likely noticing how certain processes governed by other functions within the business—such as how people are incentivized (HR) and how the budgeting process works (Finance)—limit your ability to predictably deliver better business results, and you're thinking about ways to influence change.

You've already been exploring a number of the tools you'll need to sustain successful outcomes from the now-larger team, but now you need to put in place a **vision, mission, goals and metrics** that align with the entire line of business. Lean governance is also important at this phase (we go into more detail about lean governance later on).

When setting goals, consider tying one of them directly to the cultural change you've set in motion. We've seen organizations prioritize outcomes of their transformation over the transformation process, which makes a lot of sense given how ingrained prioritizing outcomes over output has become. However, when it comes to transforming culture, we've found that until the new culture has gained a foothold, a focus on the transformation itself is needed. One way to safeguard this focus is to institute and support communities of practice.

Communities of practice

Communities of practice are communities of people who share an interest in a topic or discipline and support one another in their desire to continuously learn and improve. We often see communities of practice develop around roles such as product management, product design and software engineering, or around topics such as test automation, user research or [lean governance](#).

Lean governance

[Lean governance](#) is about ensuring that the right decisions are made with as little overhead as possible. We've found that organizations get better results when they prioritize learnings and evidence of value **over** deliverables. We've also found that a set of new processes and tools, however minimal, can be extremely useful when it comes to reinforcing and expanding this change in behavior. Key elements of our approach to lean governance are management frameworks such as objectives and key results; objectives, goals, strategies and measures; and growth boards.

Growth boards

Growth boards provide leaders with the information they need to make sound investment decisions while leaving space for teams to adapt and innovate. Whereas traditional budget planning relies on feature commitments, which often change to adapt to new information, growth boards provide the flexibility needed for modern product development.

Growth boards are an aspect of innovation accounting, which was developed by Eric Ries. A growth board is a group of cross-functional stakeholders who are accountable for making funding decisions, removing blockers and facilitating autonomy while ensuring alignment. A growth board should involve the smallest number of people possible who can get the job done while ensuring all stakeholder concerns are represented. Depending on the context, a growth board could include people from any function in the company, including central planning, IT, HR, legal, security, customer support or sales.

Establish growth boards when you want to formally introduce change into the budgeting process (as opposed to managing by exception, as you've probably been doing up to this point). This is likely to occur once you move beyond a single portfolio of products. With every team, establish a rhythm of regularly sharing key information between those responsible for developing the product and the leaders who assign their funding.

During growth board meetings, discuss:

- Product development progress along an outcome-oriented roadmap
- Results, defined by metrics, that are appropriate to each stage of the product's development
- Lessons from experiments that inform prioritization and pivot-or-persevere decisions
- How product teams and portfolios will be funded, and go/no-go investment decisions

These conversations foster high-bandwidth communication between product teams and their stakeholders. They, along with the transparency they provide, should be a natural part of the product development process.

Executive sponsors

At this level of expansion, executive sponsorship is a must. This new way of working is now attracting ample attention, and necessitates change outside the line of business. This broader change requires executive-level advocacy to be successful. Executive sponsors are most effective when incentivized on the outcomes and impact of the teams and products they sponsor.

Stage 4: Replicate – Bring change to other lines of business

Once you've achieved sustainable results in one line of business, you may feel ready to expand to others. Your primary challenge will be replicating the success you've seen so far. While some things will feel repetitive, you'll also encounter new challenges that you didn't face the first time around. A different line of business, with different stakeholders and potentially a different culture, will require that you adapt your previous techniques. Your experience with experimentation, driving for outcomes, and adapting practices along the way will serve you well as you replicate your success with new teams.

Seed teams

To increase your chances of success, seed the next line of business with experienced employees from other teams that have undergone transformation. Approach this transformation much in the same way you did with your first one: Create teams with more members who work in the new way than the old way, and provide them with protection or exceptions to process where needed. Dominant culture always wins, so create teams where the culture you want is the dominant one, and allow them time to influence those around them through demonstrated success. Over time, you'll see the new culture grow, and the number of detractors will diminish until they are far outnumbered by supporters.



FIGURE 1: Seed new teams with employees who are experienced in the practices and processes you want to adopt.

Assess your progress

As you're adapting and evolving a modern software development organization, it's important to measure team health and practices over time. You should also ensure that you're on track to achieve your desired outcomes while staying true to the principles of your organization. This measurement should be conducted by the teams themselves, at the product team and portfolio levels, as well as for any cross-cutting practices that exist in your organization. We've developed a rubric for practice leadership that we've brought to many engagements, making any modifications wherever they're needed. We deploy this rubric as a self-assessment and invite teams to come together to discuss the results, which never fail to be valuable.

Similar to how retrospectives provide space for product teams to reflect on their work and their process, these assessments are a reflection tool. However, they also provide more structure for the conversation that follows, as well as a mechanism for identifying trends over time.

FURTHER READING

- *Scaling Success with Growth Boards*
- *Don't Get Distracted by Vanity Metrics*
- *The Goal of Digital Transformation Is Outcomes, Not Engineering*
- *Tips for Scaling User-Centered Product Design*
- *4 Types of Employees You'll Need to Manage Carefully During Digital Transformation*

AGILE PRACTICE LEADERSHIP ASSESSMENT													
	Q3					Q4							
People	1	2	3	4	5	Notes	1	2	3	4	5	Notes	
Run and enable standups					X							X	
Run and foster retros				X								X	
Empower discipline practice leads to grow their practices		X							X				
Train interviewers			X									X	
Foster continuous learning & feedback		X										X	
Facilitation skills of team		X							X				
Writing high-quality job descriptions		X						X					
Maintain pipeline of talent	X							X					
Create an environment in which people are comfortable asking others for help				X								X	
Product	1	2	3	4	5	Notes	1	2	3	4	5	Notes	
Drive lean and user driven product development among teams	X								X				
Demonstrate progress through validated learnings in production			X									X	
Run effective demos to stakeholders		X							X				
Drive strong "engineering metric" performance			X										
Facilitate project stakeholder and sponsor check-in meetings	X							X					

FIGURE 2: Example of a practice health assessment

To fully assess the health of your agile organization, consider the results of any performance self-assessments against the backdrop of agile leadership criteria in conjunction with other performance indicators (e.g., DORA and product KPIs).

Assess your business impact

A practice health assessment, like the one above, tells you if you're building software better and whether you're set up to continuously improve. Make sure, though, to also regularly assess whether you're building better software for end users, and having real business impact. Getting better business results is, after all, why you've taken on transformation in the first place. Be sure to establish key metrics early, monitor them regularly, and sustain a tight feedback loop with your business partners.

Continuous improvement is critical for success

The four steps we've laid out in this paper serve as a guide for getting started, so that you can begin your transformation journey confident you know where you're headed. Even the best-laid plans can't take into account what you'll learn in the future, so you'll find that as time passes, you'll need to revisit the principles, practices, structures and processes you've established along the way. You'll also encounter challenges not addressed in these pages. Know that the strategies and techniques described here can be used in any order, many of them simultaneously. You might even invent new techniques along the way.

To achieve the flexibility and growth that enables agile software development to scale, you must be disciplined about continuous improvement and seek evidence that supports changes which lead, in turn, to better results. Set clear success criteria and intentionally assess your progress at regular intervals, at all levels of your organization. Continuous improvement is a muscle that you must train.

By consciously challenging your process and seeking improvement, you'll continue to uncover processes and techniques that propel your organization into the future.

About the authors

Becki Hyde is a Senior Product Manager and Leadership Advisor at VMware Tanzu Labs, where she works with enterprises to implement Lean Product Management, User-Centered Design, and eXtreme Programming at scale.

Jen Handler has been in product management for 15+ years, helping teams get into a rhythm of shipping valuable solutions to users and customers, and helping portfolio and line of business leaders create environments in which people deliver their best.

