



WHITEPAPER

Agile for the Enterprise

5 Steps to Help Teams Move More Effectively

IN PARTNERSHIP WITH





Learning more about Agile's competitive advantage

Anyone building complex software knows the desire is always to do it on time and under budget. That's why complex software teams of all sizes are curious about adopting Agile ways of working, which can result in faster prioritization and production.

In order to achieve those benefits, teams must follow Agile's engineering principles, which include continuous and frequent delivery, harnessing change for a competitive advantage, and efficient multi-team collaboration.

Like any professional process, Agile requires learning new skills to gain the promised benefits. And everyone needs to be on the same page. You can't have systems engineers using Agile while others responsible for overseeing planning and execution use a different method. Most often, Agile is embraced by the development team, and the challenge is addressing how the rest of the organization will adapt to the new process.

When considering Agile, we often hear folks ask questions like:

- What are we really building? What happens to the requirements?
- How do we keep everyone in the loop when we're not in the same office for daily standup meetings?
- How do we control scope and manage change?
- How do we know what the development team will deliver at the end of the sprint?

Starting an Agile Team

This paper will explore the five major challenges you and your development team might encounter as you embark on your Agile transformation.

We'll offer solutions to make Agile a success in your organization so you can reap all the benefits: better-quality products, shorter development cycles, happier customers, and bigger returns on investment.

We'll use "Agile" as an umbrella term to represent all forms of iterative development whether it's scrum, Lean Software Development, or others. And while we won't get into specific tactical intricacies of running retrospectives, writing good user stories or grooming backlogs, we will target some of the root cause obstacles and give suggestions on how to overcome them.



CHALLENGE 1

Aligning Vision with Iterations

An early challenge you'll experience is that if you have an existing Agile team, they will want to self-organize and start writing code now, and fast. They'll want to define user stories, tasks, and test cases "just enough" to create the first software iteration — a sprint in Agile terms — and deliver working code.

Getting started too quickly can be wasteful and set your team up for quick frustration.

The problem is, getting started too quickly can be wasteful and set your team up for quick frustration. It can create immediate conflict about major product attributes and how to get started on the right path. Business and technical leaders will want to know what you're building. Without that, they won't be able to communicate release plans, project scope, schedules, business models, ROI and resource plans.



Solution: Establish and Socialize Your Vision

There must be sufficient time and energy spent up front to firmly understand the product vision before distilling it into "just enough" business requirements to provide direction.

Creating a vision, documenting a product plan, and prioritizing use cases doesn't need to take the months that a Waterfall approach might take, but certainly several weeks of thoughtful customer interaction, preliminary designs and market analysis is required before getting started. The development team should participate in thinking about architecture, performance needs, user experience, platform needs, etc. However, even this front-end vision planning can apply Agile approaches using epics, fast prototyping (without writing code) and immediate customer feedback cycles to get clear, early guidance to kick off a new project.

Light documentation of this vision, clarity on who you are targeting, why customers care, and your big picture roadmap will make everyone from the CEO to the receptionist ecstatic that you have a plan.



CHALLENGE 2

Clarifying the Role of Product Owner

Another critical challenge that can cause short- and long-term angst is selecting, defining and empowering the role of product owner in your new Agile process. Product owners are responsible for being the voice of the customer, the evangelist and decision-maker, and the perfect blend of business acumen and technical savvy.

The product owner is a critical role to drive priorities, approve software releases and be a liaison between the product development team and the rest of the company, as well as the market.

Additionally, while the product owner role is heavily ingrained in development, someone must be equally entrenched in a more business-focused, product management role doing market and customer research.

Selecting the wrong people or incorrectly defining these roles introduces risk into your entire Agile product development process.



Solution: Think About Responsibilities

You should think about the role of product owner more as a set of activities, interactions and desired outcomes rather than a job title, and structure your team and responsibilities accordingly.

One of the main responsibilities of a product owner is spending a significant amount of time directly with the development team.

The product owner participates in every iteration review (often multiple meetings per week), writes and reviews use cases, helps write and approve test cases, and is available to review and approve software releases. This is a very hands-on role that requires serious time and commitment.

Challenge 2: Clarifying the Role of Product Owner

Getting the product owner and the product manager roles in a room to work out how they will work together and how decisions will be made at the very tactical level is a key step to success.

Our suggestion is to have the product manager participate in planning meetings, agree on priorities and implementation, then allow a more technical product owner to drive day-to-day decisions, write use cases and approve test cases. Have the two roles sync back up for software releases and fix any conflicts with the next sprint.

But take heed, when creating complex products and systems, it's not advisable for one person to be responsible for all the activities required by both roles.



CHALLENGE 3

Not Building in Real Customer Feedback Loops

A major tenet from the Agile Manifesto is, “Our highest priority is to satisfy the customer.”

However, let’s be clear: The product owner is not the customer; marketing is not the customer; and the CEO is not the customer.

Only the customer is the customer. This may sound obvious, but this is by far the biggest challenge to Agile development teams working on market-focused products.

When Agile first got traction with internal development efforts, bringing customers directly into the Agile process was relatively straightforward. Simply take a completed iteration down the hall and sit with the “customer” to get their feedback. However, as Agile spreads to more open-market solutions, gaining real customer feedback in a timely manner is more difficult. It’s particularly challenging for a new project that doesn’t have any paying customers yet, and even more challenging for consumer-based products, where the “customer” often feels like a mass market of people.

There are several reasons why teams find it challenging to bring in real customers during the Agile development process:

1. The perception that these activities will slow the team down.
2. The input is vague and often ignored.
3. Uncertainty of who the customer really is.
4. They don’t know how, or they try to rely on traditional surveys or focus groups.

Because of these challenges, it’s easy for developers, or even product owners, to take shortcuts and use personal opinion statements to drive product decisions, rather than build in real customer feedback.

Challenge 3: Not Building in Customer Feedback Loops



Solution: Plan for Rapid Customer Insight

To be truly Agile, it's critical to bring customers into your efforts at the right points and with the right methods. While gaining real customer insight throughout Agile planning and development may seem challenging, it doesn't need to be.

We use three simple and equally important steps to gain rapid customer insight that supports Agile development efforts. These steps are:

1 ACCESS

You must find and identify a set of target customers that you can rely on to provide accurate, timely insight. These are often early adopter customers that will not only share their insight but want to be part of your success. Successful Agile requires developing a well-maintained customer panel or advisory board.

2 LISTEN

Once you have direct and rapid access to customers, you must build skills to actively listen to them. This isn't running a focus group, launching a survey or asking them what they want. Although these methods can also be used, having high-quality interaction with your customers — in person or through collaboration tools — is critical, as is probing customers for real needs, problems, desires and objective feedback. Listening also means sharing designs early to learn what your customers are thinking, how they would prioritize elements of your solution and the trade-offs they are considering.

3 COMMUNICATE

Disclose learned insight into your development efforts through clear and prioritized use cases, explain the relative value of each feature and build test cases that reflect how your customers would want to experience your product.



CHALLENGE 4

Developing a “WaterScrumFall” Process

As shared earlier, management needs a roadmap, a schedule, a vision document, a plan. This is one of the main reasons that many companies either overtly or covertly create a hybrid of Waterfall and Agile. They use Waterfall to clarify the front end to develop a plan, and then allow the development team to take over and use an Agile approach. Once the product is near market release, the team will attempt to go back to the plan developed under Waterfall thinking.

While this hybrid process can work, it creates great strain on the organization due to the management team following one process and the development team using different process philosophies, terms, and metrics. In Waterfall, once a plan is baked and approved, there is an expectation that the plan will be followed and delivered upon, even if the development team is using Agile to execute.

Now I’m going to say it: “But that’s not really Agile.” True Agile requires the plan to be consistently reprioritized and revised. We see

this approach so often that we’ve heard many describe it as “WaterScrumFall.” It’s really business as usual with a traditional process of defining a complete product up front and then the development team using an internal Agile process to conduct the work break-down process to deliver code. But often the real testing and development doesn’t even start until testing of the expected deliverable starts. This is way too late to leverage the power of “agility” in software development.

The bottom line: Any Agile approach used by the development team must support all business needs and address all stakeholder concerns.

Challenge 4: Developing a “WaterScrumFall” Process



Solution: Align Sprints to Roadmap Milestones

True Agile requires that management, marketing, operations, and other functions are aligned with the principles of Agile development. Agile evangelists must acknowledge the needs of business leaders and other departments, and these other groups must acknowledge the methods and benefits of Agile.

In more concrete terms, product roadmap milestones and market releases developed in the Waterfall model must be aligned completely with Agile sprints and software releases.

If the development team is practicing Agile, it must create deliverables that track to the plan and communicate early and regularly on what’s getting completed, and how it reflects on the roadmap.

To guide the development team’s iterative approach, the marketing and sales teams must be clear on what customers deem most important and how market dynamics are impacting solution requirements.

Progress and product deliverables must also be communicated in both Agile and business teams. For example, use cases and tasks must be translated to the promised features, and business models must be broken down into user stories.

The bottom line: Product deliverables must be communicated early and regularly across both Agile and all other relevant business stakeholders and teams.

CHALLENGE 5

Losing the Forest for the Trees

If you've already solved the first four challenges discussed, the early stages of a new Agile project should be running smoothly. Your team is working on user stories, test cases and building features. The vision and plan have (hopefully) been established and everyone's excited about being outrageously successful. You've also comfortably tackled some of the early platform and architecture efforts.

As Agile hums along, however, the backlog increases, new ideas come into the mix, bugs stack up and the development team starts getting tired and frustrated. Progress appears to be slowing down since more time is spent on bugs, design changes, and minor enhancements.

At this point, it seems easier to focus on what can get done over what should or must get done. You may start wedging small features and incremental tweaks into sprints while bigger, more challenging and more valuable problems are not addressed because these bigger efforts don't allow room to fix bugs and finish features.

Decisions get tougher and frustrations set in. Your management team may even start thinking that Agile isn't working, since the plan is not being delivered upon.



Solution: **Go Back to Basics**

At this point it's more important than ever to go back to the basics: clarify the vision, listen to real customer input and focus on the main thing you set out to deliver.

What are the features, user stories, use case, and other attributes that you must get right to be successful in the marketplace? As your solution gets close to delivery, Agile can't be a philosophical software development process, but a business process for delivering greater value to customers and competitive offerings to the marketplace.

Challenge 5: Losing the Forest for the Trees

Use your Agile skills to make the tough decisions: cut less important features that aren't complete, ignore seemingly critical (but not important) bugs and re-energize the team on product attributes that your customers care about most.

These tough decisions obviously can't wait until the final pre-launch but must consistently be made through the entire development process.

While nothing provides more satisfaction than a complete product that does everything you want it to do, when the schedule conflicts with completeness (and it always will), err on delivering a solid solution that does less. Then, get it into the hands of real customers, learn, iterate and succeed.





In Summary

Now you have real solutions to address the challenges you'll face in adopting Agile processes. This will enable you and your teams to start with a clear vision of the problem you want to solve for customers, clarify who the customers are, understand what they would really value in a system and outline a solution to get started.

You'll be able to identify a product owner and product manager, and ensure they develop a clear working relationship to make decisions and communicate progress. You'll create successful customer feedback loops to consistently bring in customer insight to guide the team's designs and priorities, and to validate your solutions.

While Agile is new to your company, you can help translate Agile methods, metrics and terms to language that is clear to your

business leaders. Furthermore, you can get leadership comfortable with your progress and help them see the power in staying flexible throughout the development effort as new information surfaces and customer learning evolves.

And when the going gets tough, your Agile skills can help refocus efforts on the most valuable features to give your product the best chance in the market.

Interested in learning more about how Agile methods will help your team move more efficiently?

Visit jamasoftware.com/solutions/requirements-management to learn how Jama Connect can streamline Agile methods and requirements gathering for your teams.



ABOUT JAMA SOFTWARE

Jama Software® is focused on maximizing innovation success in multidisciplinary engineering organizations. Numerous firsts for humanity in fields such as fuel cells, electrification, space, software-defined vehicles, surgical robotics, and more all rely on Jama Connect® requirements management software to minimize the risk of defects, rework, cost overruns, and recalls. Using Jama Connect, engineering organizations can now intelligently manage the development process by leveraging Live Traceability™ across best-of-breed tools to measurably improve outcomes. Our rapidly growing customer base spans the automotive, medical device, life sciences, semiconductor, aerospace & defense, industrial manufacturing, consumer electronics, financial services, and insurance industries. To learn more, please visit us at jamasoftware.com.



ABOUT THE AUTHOR

Dorian Simpson has led the successful development and launch of innovative products and services for over 20 years and has always been at the forefront of new digital media — from leading the development of the world's first 500-channel digital satellite system, to taking the world into Digital Video Recording and delivering the latest mobile applications. Dorian received a BSEE from Northwestern University, an MBA from USD and lives in Portland, OR. You can email him at: dsimpson@planninginnovations.com.

Auxilium (formerly Planning Innovations) was founded in 2002 to enable clients to deliver better-targeted consumer solutions — faster and with greater success — using rapid customer insight techniques. Their services include highly rated programs in innovation leadership, product definition, product strategy development, and process assessment. Learn more at auxilium-inc.com.