Agile for the Enterprise
The 5 Steps to Successful Adoption

Agile for everyone. Yes, it can be done. As the Agile Manifesto methodology matures, the values it offers engineering teams remain, but its principles, including continuous and frequent delivery, harnessing change for competitive advantage, efficient multi-team collaboration—and the resulting benefits of faster prioritization and production—have many business teams curious about adapting Agile for their ways of working.

Like any professional process, Agile requires learning some new skills to gain the promised benefits. You can't have Systems Engineers working Agile while the Product Planners, Product Definers, Product Owners, Product/ Project Managers and others responsible for overseeing planning and execution are left behind. Most often, Agile has been embraced by the development team and the challenge is addressing how the rest of the organization will adapt to the new process.

We often hear 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?**

This white paper addresses five of the major challenges that we've seen lead to Agile failure, as well as solutions to make Agile hum for your organization. We'll use "Agile" as an umbrella term to represent all forms of iterative development whether it is SCRUM, Lean Software Development or others. We also won't get into the specific tactical challenges of running "retrospectives," writing good "user stories," or "grooming backlogs," but we will target some of the root cause challenges with suggestions on how to be successful.

Let's Start an Agile Team

To set the stage, let's visualize an Agile team getting started. Your senior team has heard all about Agile and wants to gain from all the benefits–better products, shorter development cycles, happy customers and bigger returns. You, as the Agile evangelist, have been selected to lead this effort. You also have a new project just getting started. The project is going to be a new home monitoring system that will cut down electric usage in the average home by 93%. You can picture building your Agile development process and sharing your success with the rest of the company until everyone is bathing in Agile goodness.

Not only will you demonstrate the power of Agile, but solve many of the world's problems in one fell swoop. We'll further assume that your team is already using a collaborative requirements management tool and is skilled at writing good user stories and use cases, creating test cases, tracking bugs and all the other fundamentals of good development.

Now let's look at the challenges and review them in about the same order as you, your team and those interacting with them might experience them. At the end, we'll come back to our home monitoring system example and the possible results based on whether we were able to overcome these challenges.
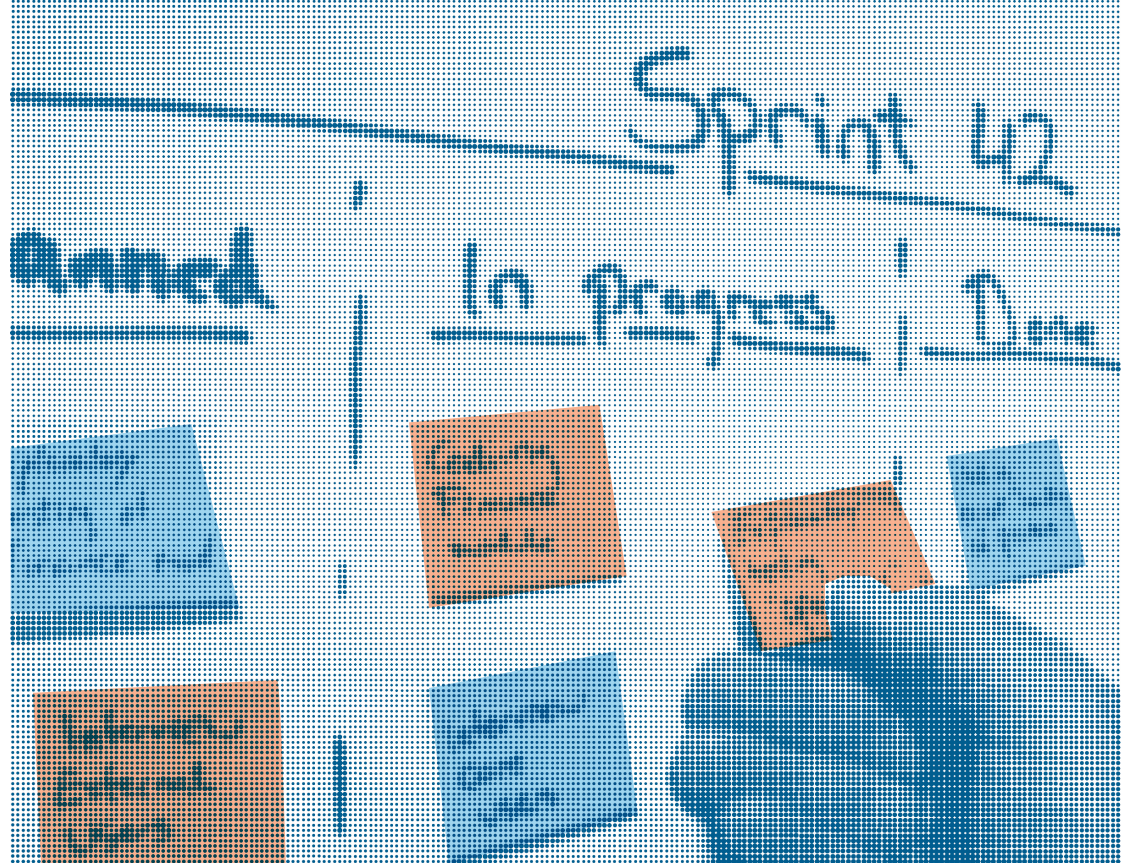
Challenge One
Aligning Vision with Iterations

An early challenge you'll experience is that the existing Agile team will want to self-organize and start writing code now… 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. The Product Owner (which we'll discuss next) might state, "We know that customers will want to monitor every outlet and appliance in the house, let's start with building a database schema that allows users to enter a list of these."

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

While this is not inherently bad, getting started too quickly can be wasteful and set your team up for quick frustration, creating immediate conflict with other ideas about the major product attributes and how to get started on the right path. Management, and even some internal technical leaders such as Architects and Product Managers, will scream, "Wait! What are you building?" At which point the Agile development team might respond, "We're just getting started… we'll refine it along the way."

Unfortunately, this is little comfort for those that must communicate release plans, project scope, schedules, business models, ROI and resource plans.

The Solution

Even the worst car drivers will have a general understanding of where they are going before making their first turn out of the driveway. There must be sufficient time and energy spent up front to gain a solid grounding in 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 of course 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.

© Jama Software, Inc
www.jamasoftware.com

Try Jama for Free
Request a Quote

Ask a Question
Schedule a Demo

Call us direct:
503.922.1058

Page No.
2

Challenge Two
Clarifying the Role of Product Owner

Another critical challenge that can cause short- and long-term angst is in selecting, defining, and empowering the role of Product Owner in your new Agile process. Let's accept that this is one tough role. 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.

and desired outcomes rather than a job title, and structure your team and responsibilities accordingly. As a Product Owner, one of the main roles 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.

> The role of Product Owner is indeed challenging. You should think about it more as a set of activities, interactions and desired outcomes rather than a job title.

Should you select a Product Manager that has previously been the product champion and tradeshow expert– but doesn't know Agile software development? Perhaps you should select a Project Manager or Program Manager that understands software development and focus groups. Or, perhaps you should nominate a solid, innovative architect that can design a 20 billion-user system?

Whatever the choice, the Product Owner is a critical role to drive priorities, approve software releases, and be a liaison between development and the rest of the company… and often the market. Selecting the wrong person or incorrectly defining the Product Owner role leaves your Agile team limping along, or worst, at the whims of a control freak bent on driving personal opinions into the product.

The Solution

The role of Product Owner is indeed challenging. You should think about it more as a set of activities, interactions

If someone is this engaged with the development team, who is doing all that important market and customer work? It has be someone, or expect to fail. Someone, such as a Product Manager or Product Marketing Manager must take this more business-focused role and be the yin to the Product Owner's yang. Getting these two 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.

One tip–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 them sync back up for software releases, and then fix any conflicts with the next sprint. If one person is responsible for all of these activities, only a superhero will be successful except on small projects or in much-defined markets.

© Jama Software, Inc
www.jamasoftware.com

Try Jama for Free
Request a Quote

Ask a Question
Schedule a Demo

Call us direct:
503.922.1058

Page No.
3

Challenge Three
Not Building in Real Customer Feedback Loops

A major tenet of Agile 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. The people in marketing are NOT the customer. The CEO is NOT the customer. The only person that is the customer is... well... the customer. This may sound like a "duh" moment, but this is by far the biggest challenge to Agile development teams working on market-focused products.

Bring customers in at the right points, with the right methods.

When Agile first got traction with internal development efforts, bringing customers directly into the Agile process was relatively straightforward. Just take a completed iteration down the hall and sit down 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, and is 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 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 opinions statements such as, "I'd want this feature," "It should work like this," or "The customer will need this," to drive product decisions, rather than build in real customer feedback.

The Solution

To be truly Agile, it is 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 support Agile development efforts. These steps are:

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.

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 though 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 tradeoffs they are considering.

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 Four
Developing a "WaterScrumFall" Process

As shared earlier, management needs a roadmap, a schedule, a vision document, a plan. "But that's not Agile!" says the Agile team. 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 their Agile approach. Once the product is near market release, the team will attempt go back to the plan developed under Waterfall thinking. This often includes applying the launch-readiness criteria developed in the front end planning and gets a quick response from

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

management, "Hey... this isn't what we agreed to!" to which the development team responds, "Hey, it's Agile." 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 real 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. Let the blame games begin!

The Solution

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 provide early warning of what is really 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 to guide Agile efforts with every Agile sprint. Communication of progress and product deliverables must also be spoken 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: Any Agile approach used by the development team must support all business needs and address all stakeholder concerns.

© Jama Software, Inc
www.jamasoftware.com

Try Jama for Free
Request a Quote

Ask a Question
Schedule a Demo

Call us direct:
503.922.1058

Page No.
5

Challenge Five
Losing the Forest for the Trees

In the early stages of a new Agile project, all is good. Your team is working on user stories, test cases, building features, and happily coding along. The vision and plan has (hopefully) been established and everyone's excited about being outrageously successful. You've also comfortably tackled some of the early platform and architecture efforts.

Now if you've already solved the first four challenges discussed, this challenge–Losing the Forest for the Trees–will be much less of a challenge. But as Agile hums along, 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 in small features and incremental tweaks into sprints while bigger, more challenging– and more valuable–problems can't be addressed since 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 for you since the plan is not being delivered upon.

The Solution

As one of my client's executives put it (and I'm sure he borrowed it from somewhere), "You need to keep the main thing, the main thing." 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 what the "main thing" is. What are the features, user stories, use cases, 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.

Use your Agile skills to make the tough decisions to cut less important features that aren't complete, ignore seemingly critical (but not important) bugs and re-energize the team on those 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

© Jama Software, Inc
www.jamasoftware.com

Try Jama for Free
Request a Quote

Ask a Question
Schedule a Demo

Call us direct:
503.922.1058

Page No.
6

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

Let's review how your team did in delivering on the new home monitoring system that will cut down electric usage in the average home by 93% that we discussed in the introduction. Starting with a clear vision of the problem you wanted to solve for customers, you clarified who the customers are, what they would really value in this system and outlined a solution to get started. As the Product Owner, you also developed clear working relationships with other marketing leaders to make decisions and communicate progress. You also created 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 helped translate Agile methods, metrics and terms to language that is clear to your business leaders. You also made them comfortable with your progress and helped them see the power in staying flexible throughout the development effort as new information arose and customer learning evolved.

When the going got tough, you used your Agile skills to refocus efforts on the most valuable features to give your product the best chance in the market. While many more iterations remain to fully achieve your vision, customers love your home monitoring solution, are enjoying vast energy savings and have peace-of-mind knowing they are saving the planet. Success is imminent. Agile is good. I think we can agree on that. If you're considering adopting Agile methods, please consider these five challenges as they relate to the planning phase of development.

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, a MBA from USD and lives in Portland, OR. You can email him at: dsimpson@planninginnovations.com.

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 www.planninginnovations.com.

About Jama Software

Jama Software is the definitive system of record and action for product development. The company's modern requirements and test management solution helps enterprises accelerate development time, mitigate risk, slash complexity and verify regulatory compliance. More than 650 product-centric organizations, including NASA, Boeing and Caterpillar use Jama to modernize their process for bringing complex products to market. The company is headquartered in Portland, Oregon. For more information, visit www.jamasoftware.com.

© Jama Software, Inc
www.jamasoftware.com

Try Jama for Free
Request a Quote

Ask a Question
Schedule a Demo

Call us direct:
503.922.1058

Page No.
7