Embracing Agile

Ten Tips for Understanding Agile Development
and Making it Work for You

WHITEPAPER

APPLAUSE
WINNING APPS GET IT

Introduction

Since the creation of the Agile Manifesto (excerpt right) the Agile methodology has been talked about, taught, tweaked, adapted, reviewed, rejected, heralded and ultimately worked its way deep into the software development culture.

There have been multiple "spin-offs" of Agile since the manifesto was first published, including Extreme Programming, SCRUM, DSDM, Adaptive Software Development, Crystal, Feature-Driven Development, Pragmatic Programming and Agile-Fall, among others. Some of these have come and gone while others have stuck around. And though they differ in subtle ways, they all hold the same foundational values.

This whitepaper will walk you through some key ideas behind successful Agile practices, some challenges you're likely to run into and how to make the most of your Agile efforts.

Agile Manifesto
We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals & interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

1. Define and Understand At All Levels

Though Waterfall is the methodology more commonly associated with planning, understanding and defining work and project goals is no less important in Agile.

When embarking on an Agile project, it is extremely important to identify a product owner. This person should understand the project's big picture, know how sprints fit together and act as the go between for the delivery teams and upper management. Agile has a lot of moving pieces and someone needs to understand how they all fit together.

Once a product owner is in place, they should take the lead on establishing overall product goals with the delivery team and helping team members understand the objectives driving those goals. This gives teams a clear idea of what they are working toward, and why and will help them plan their own roadmaps.

Throughout the project, there may be various business factors that effect the delivery teams. The competitive landscape, regulatory issues, customer satisfaction, service level agreements, stock price, missed sales forecasts and other outside factors can force a team to adjust their sprint order and reprioritize features. The ability of a team to manage change and match shifting business goals is a major benefit of Agile. Understanding that changes may (and most likely will) happen helps teams stay on their toes and move quickly if a change does occur. If all members of the team understand the overall project goals, adapting should be easy.

On the Sprint Level

Establishing the goals of a sprint at the beginning gives teams a clear idea of what they are working toward with that particular feature. Well-understood goals allow teams to know when a sprint is truly done (another key component of Agile) and protects against feature creep.

Agile development is the practice of completing small, set portions of a product and launching them independently. If you don't adequately define "done" – including what needs to be tested and what documentation is needed – a sprint may be "completed" before it is really 100% done or veer off course into unrelated features. For instance, teams that ignore bugs and move to another iteration are not really completing sprints. Instead, they are leaving loose ends to be tied up later, which isn't in the Agile spirit. Defining "done" means your team has a clear understanding of what needs to be completed with every sprint – no more, no less.

2. Know the Big Picture

Focusing on the "done-ness" of individual sprints makes it easy to forget that the overall final product. While each sprint is independently completed and released on its own, how do all the pieces work together? When more than one piece of the puzzle comes together, it's important to remember routine responsibilities like regression testing, load testing and overall security testing. Occasionally focusing on the product as a whole will help ensure you produce a high quality application overall, not just high quality individual features.

Regression Testing

At the end of each sprint, it is important to test that product against pre-existing pieces. While the sprint may be thoroughly tested, it is possible for this new release to adversely affect a previously completed component. Never forget that in the end, all sprints are working toward one common goal – a successful, high quality product.

Load Testing

At what point will your application's performance begin to degrade? How many concurrent users can it support? Where are the bottlenecks between your code base, database, CDN and load balancers? Without answers to these questions, your testing could be all for naught. Load testing the completed end product is more important than load testing each sprint. If your software can't hold up under stress, it doesn't matter how agile your methods are.

Security Testing

It is extremely important to do security testing not only during sprints, but also on the overall product. You want to make sure hackers can't sneak in through the gaps after the software has been stitched together. You also want to be sure that data isn't leaking from some unaccounted-for area that didn't make it into any of the sprints. Be particularly careful that adequate security testing is done in the Agile world of less documentation.

3. Incorporate Testing Early

Despite the title of "quality assurance," testers can never truly assure quality, nor can they find every bug. The real goal of testing should be to improve the software. This is a widely held view of testing in the Agile world.

In an Agile environment, testers provide valuable information throughout the development cycle, as opposed to issuing a ‘pass’ or ‘fail’ grade at the end.

If Agile is to succeed, testing must become a central pillar of the development process. Testers should be full members of Agile teams, working on projects and individual sprints from the very beginning. Quality is everyone’s responsibility and incorporating testers right away helps spot logic and quality issues early, which in turn prevents major problems down the line. If you don’t fully understand the importance and role of testing, you will be missing a major component of Agile.

> “Testing is not a phase on Agile teams, testing is a way of life. Agile teams test continuously. It’s the only way to ensure that the features implemented during a given iteration or sprint are actually done.”
>
> -Elisabeth Hendrickson
> Founder, Quality Tree

4. Capturing Meaningful Data

Compiling detailed reports that show the number of test cases written (i.e. how many have been run, failure rate, severity levels, etc.) is certainly valid data to the testing manager, but who else in your organization will care? Remember, Agile is all about minimizing paperwork and only spending time on documentation – including data – that is absolutely necessary. As testing expert Michael Bolton said:

“When you enslave numbers, they eventually rise up, revolt and enslave you. These organizations spend so much time collecting the data and talking about it and justifying it and trying to duck blame that they don’t seem to have time to do anything about the actual problems, which generally fall into two categories. One: the organizations are trying to do more work than they can handle with the approaches they're using. Two: they're not listening to people that matter – neither to their customers, nor to their own front-line staff, many of whom are closest to the customers.

When your car is about to go off a cliff, it's a weird time to be thinking about gas mileage and drag coefficients; better to take the right control action – look out the window and steer or use the brake until you're back on course."

Focus on data that is helpful to several departments, further the efforts of fast moving Agile teams and help produce better applications.

5. Common Concerns

Making such a radical shift can understandably be nerve-wrecking and management is sure to have some questions and concerns. Here are a few common concerns about Agile, and why you shouldn't be worried.

Lack of Planning

A large part of Waterfall development is spent planning. Switching to Agile can leave teams feeling crunched for time or lost. As we discussed in the first tip, though, agile isn't a free-for-all.

Planning in Agile is simply segmented. Once product goals are set, a team picks a feature to work on and plans for that feature and that feature alone. As the teams move through the project as a whole, they may identify aspects that were inadvertently left out of planning in the first few sprints. The nature of Agile means missteps affect only small portions of the project and corrections can be made quickly for future sprints.

Lack of Documentation

A key component of Agile is doing away with unnecessary, clunky documentation. This, however, does not mean getting rid of all documentation. Many teams use index cards or a gridded whiteboard to represent sprint requirements, goals, step order and to show what each team member is working on at a given time. Documentation is simply leaner.

A sprint is not complete until all necessary product documentation is written. This traditional documentation isn't forgotten or done away with, it's just completed in parts rather than at the end of an entire project.

Daily Meeting Hassle

While the prospect of a daily meeting may seem time consuming, the purpose of Scrum is to be a quick check-in and organization meeting. If you honor the purpose of Scrum you will find it is more useful then hassle.

6. Keep Scrum Effective

Far from an Agile prerequisite, daily Scrum meetings have proven to be extremely beneficial in the long run by helping teams remain focused, energized and coordinated. Scrums keep team members responsible to each other. Each member knows what the others are working on and where their piece fits into the puzzle (or conversely, where their work is making others wait). Transparency and connectivity are staples of an efficient team and regular scrum standups are a good way to foster those traits.

Too often, however, daily Scrums become weekly meetings, which in turn become monthly meetings. If meeting less often works for your team – great! But the Scrum model isn't effective if meetings are inconsistent, too long or involve unnecessary information. It helps to appoint a Scrum Master who will lead the meetings and ensure teams adhere to a regular schedule and time limit.

7. Heads-Up: Switching is Hard

Do not assume that switching to Agile will be a quick, easy transition. Product owners, business analysts, development and testing departments all need time to learn, understand and work the kinks out of this new company approach. This takes time.

When switching, be sure the Agile teams – and the executives overseeing their work – fully embrace the practice. This will help you avoid the pitfalls of "Fake Agile." Fake Agile is a term coined by Software Tree Company founder Elisabeth Hendrickson, "It removes all the external controls that made traditional practices work without instilling any of the team-centered disciplines that make agile work. The end result is usually worse quality, slower and an organization that blames 'agile' for the mess." While you should be aware of Fake Agile, don't confuse it with Agile-Fall or other hybrid practices.

8. Making Agile Work for You

Several off-shoots of Agile have emerged in recent years. These off-shoots – like Agile-Fall or Scrumfall – are the byproducts of organizations attempting to adopt Agile practices and ultimately settling on a hybrid approach that works best for them. Hybrids aren't bad, they are just another option for teams to get the job done. The important thing is to find a method that works for your specific situation. As long as the main goal of producing better products quicker and with fewer resources is maintained, the spirit of Agile is intact.

If you are unsure if Agile is right for your company, start with a project that isn't too complex or mission critical. Starting with a smaller, less important project will allow you to see how Agile works within your organization while minimizing any potentially negative business impacts.

> "There's no shame in [hybrid methods] if that's what works. When you're going through a transition from Waterfall to Agile, that may be the best thing as opposed to a sudden lever-pull one day where you show up and your desk is next to someone else with no walls and there's a stack of sticky notes and markers on your chair with an email to report to your first standup in 30 minutes."
>
> -Jon Bach
> Director of Live Site Quality, eBay

9. Being Agile Without Burnout

The short release cycles of Agile development can often place teams under tight time constraints. Leveraging outside help when possible is a good way to stay on schedule and give teams a much needed break. Weekends are typically a period of low activity, and this is a good thing – people need their rest to maintain productivity. However, this two-day gap need not go to waste.

While development likely needs to stay in-house, turning to an outside source for extra testing is a way to ease the pressure on your team and get fresh eyes on the product. Outside testing options, like crowdsourced testing, can help teams make the most of natural gaps. By leveraging a crowdsourced testing solution, teams can deliver software for testing on Friday and receive results by Monday morning, compressing the development cycle without adding pressure to in-house staff. Crowdsourced testing is a better option than traditional outsourcing because crowdsourced models offer more flexibility, scalability and feedback that includes valuable real world metrics (some of that focused, important data we were talking about earlier).

10. Agile Isn't for Everyone

There is nothing wrong with other methodologies. If you think your team, department and, ultimately, company will benefit from Agile then give it a shot. But if everything is working smoothly, teams are happy and successful products are being launch on time and on budget, Agile may be an unnecessary, burdensome change. Similarly, if thorough documentation and in depth accountability is necessary for your industry, Agile could be down-right detrimental.

> "Adopting Agile won't fix a dysfunctional team and it won't help an organization to learn to accept its limitations and work within them."
>
> -Noah Sussman
>
> Test Architect, Etsy

Before switching to Agile, take a moment to review current practices, flag the major issues you're hoping to correct, study and fully understand how Agile development works and objectively evaluate if Agile will help your company. Take your time switching and be sure all the key players are on-board.

If you try Agile and it simply isn't working, revert to the old method. Remember, it's about embracing what works for your company, not adhering to a development methodology for the name alone.

About Applause

Applause is leading the app quality revolution by enabling companies to deliver digital experiences that win - from web to mobile to wearables and beyond. By combining in-the-wild testing services, software tools and analytics, Applause helps companies achieve the 360° app quality™ they need to thrive in the modern apps economy. Thousands of companies – including Google, Fox, Amazon, Box, Concur and Runkeeper – choose Applause to launch apps that delight their users. Learn more at www.applause.com.