

APPLAUSE^o

WHITE PAPER

ESSENTIAL GUIDE TO AGILE TESTING

TABLE OF CONTENTS

INTRODUCTION.....	3
WHAT AGILE TESTING SHOULD LOOK LIKE	3
WHAT AGILE TESTING ACTUALLY LOOKS LIKE TODAY	4
OVERCOMING BARRIERS TO EFFECTIVE AGILE TESTING.....	5
Time	5
Money	6
Planning	6
BEST PRACTICES FOR AGILE TESTING	7
Partner with the Business Side	7
Find the Right Balance Between Automated and Manual Testing	7
Expand Your Capabilities by Crowdfunding	7
CONCLUSION	8

INTRODUCTION

The speed and adaptability of agile development is no longer a competitive advantage. It is a requirement. [According to Atlassian](#), 80% of all software organizations now practice agile.

Gartner Analysts identify two key trends driving this shift in a recent [report](#):



Firstly, related products and services will be increasingly digital. Secondly, business capabilities will have to operate in a more dynamic environment driven by regional and global economics, regulatory mandates and increasingly demanding, technology-savvy customers.

DAVID NORTON & MIKE WEST, GARTNER ANALYSTS

In spite of this need for agile, running an effective agile development process remains very rare.

The rapid nature of agile development often forces developers to choose between adding functionality or testing for quality. As a result, issues slip through cracks only to reappear later in the development cycle when there is no time to address them. When this happens, carefully planned timelines are derailed and product shipments are delayed. Or worse, your customers will find a bug instead of your developers.

But you can avoid disasters like this by adjusting your approach to testing within your existing agile development framework.

WHAT AGILE TESTING SHOULD LOOK LIKE

In an agile process, testers should provide valuable information throughout the development cycle, as opposed to issuing a 'pass' or 'fail' grade at the very end. Whether you are practicing true agile or [Wagile](#), if your effort is to succeed, testing should be a central pillar supporting the entire development process.

Testing and development should be performed in concert with each other; with developers focused on executing continuous small projects, and testers focusing on testing those projects at every step.

Each project and corresponding test should be considered one sprint. The quicker your sprint cadence, the more adaptive, or “agile,” your team can be when adding new functionality or responding to unexpected mistakes and customer feedback.

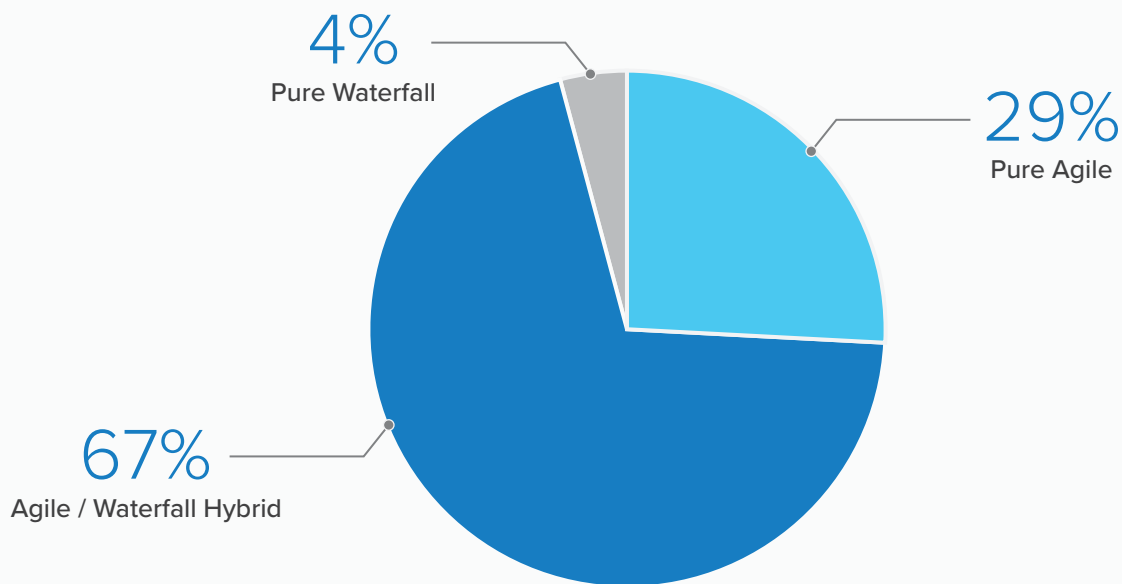
At the same time, a quicker sprint cadence also requires a quicker testing cadence to keep up - and this is where so many companies get lost. In order to run an effective agile development process, testing must keep pace with development every step of the way.

A good analogy is thinking of each individual sprint like a building block. Pairing agile testing with agile development in each sprint means that, as you construct a building, every block is structurally sound and inspected to ensure high quality along the way. When you’ve been building with high quality material at each step of construction, the onerous task of testing the entire project upon completion is much more manageable.

WHAT AGILE TESTING ACTUALLY LOOKS LIKE TODAY

According to a survey conducted by voke, three-quarters of enterprises and over half of tech firms do not have QA working concurrently with developers during sprints. This means that development and testing are executed on a separate cadence, with testing usually happening later and less often.

ONLY 29% OF QA PROFESSIONALS BELIEVE THEIR COMPANIES EXECUTE PURE AGILE DEVELOPMENT

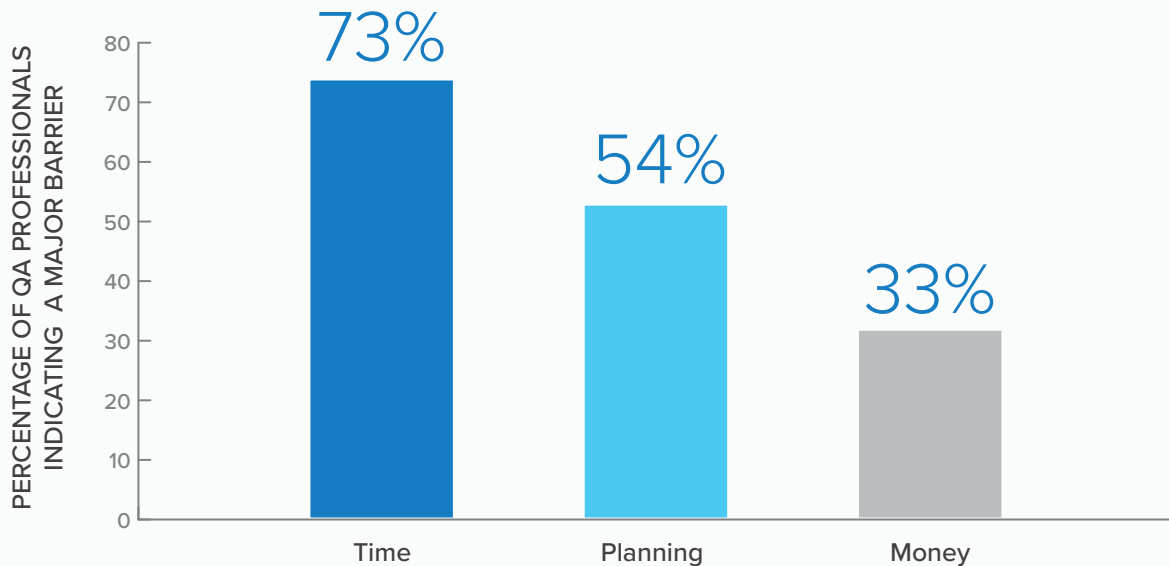


Neglecting to test after each sprint may not seem like a big deal. You’ll always be able to catch any issues from this sprint when you test at the end the next sprint, right? That may be so, but playing this game shrinks your margin for error in the future, and that will inevitably catch up with you.

Forcing two sprints worth of testing into the timeframe for one sprint will compromise even the most thorough tester's ability. Then, when the focus of testing changes for the next sprint, any issues that evaded your testing team will remain lurking in your code for the indefinite future.

These are the “zombies” that show up when you are preparing to release, and depending on how severe of an issue, have the ability to cancel your release altogether. The moral of the story: In order to prevent zombies from ruining your product, it's critical to find a sprint cadence that also accommodates your testing capabilities.

OVERCOMING BARRIERS TO EFFECTIVE AGILE TESTING



Source: uTest Community of Digital Experts, n=257

Why wouldn't you test after every sprint - especially if it is so abundantly clear why this practice is critical to success? Here are some of the most frequently mentioned barriers:

TIME

It is arguably the most valuable resource in the world, and [46% of developers](#) claim they don't have enough time to test as much as they should. However, [developers also have cited](#) that their last major software defect required an average of 20 developer hours to fix.

While testing after each sprint requires testing at a higher frequency, it also requires testing at much shorter durations. These brief durations should add up to much less than 20 hours; not to mention each of those hours of proactive testing will be much less stressful and cumbersome than each hour of executing on an unexpected and high-stakes hotfix.

It may seem like you don't have the time to test. However, it is more true that you don't have the time not to.

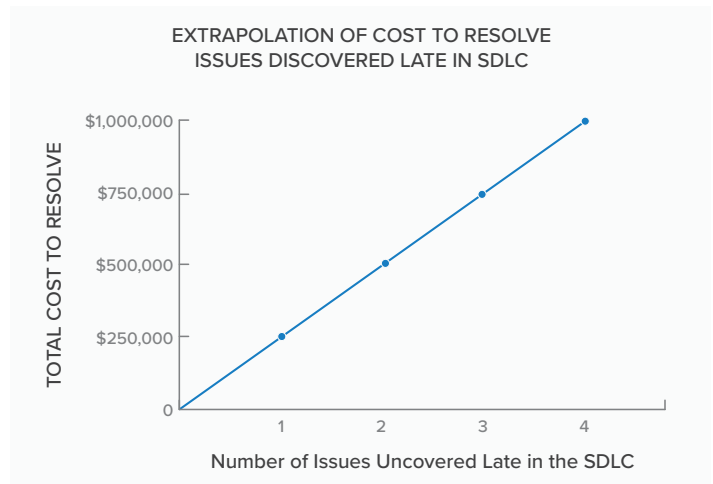
MONEY

Taking an economic view of testing is always the right approach, but make sure that you aren't missing a major part of the formula.

The common-sense attitude is, "we incur overhead every time we test so let's minimize that overhead by testing less often. Since we have to test at the end anyway, let's just defer testing to that point."

What is missing in this formula is the cost of the previously mentioned "zombies" lurking in the dark, waiting to show up when you are getting ready to launch. **That cost is [estimated by developers](#) to be an average amount of \$250,000 per issue.** Similar to how

investing time up-front will minimize hours of frantic fixes down the road, incurring small amounts of overhead each sprint saves you from becoming penny wise and pound foolish at the time of launch.



PLANNING

Then, there is the infamous "[Planning Fallacy](#)" that plagues us all. Ever since this phenomenon was first proposed by psychologists in 1979, it has explained why people tend to disregard negative past experience when estimating future requirements to complete tasks.

It is human nature to view your own future with unwarranted optimism, while still viewing everyone else's with a healthy level of skepticism, but don't let this tendency tank your QA effort.

Yes, it is within the realm of possibility that developers will consistently code well enough to render any testing after a sprint a waste of time, money, and energy. But past experience indicates that is not likely. Developers who felt their companies allotted sufficient time to pre-release testing indicated spending less than half as much time resolving bugs later in the development cycle.

Plan for mistakes, and plan for bad ones too. They have happened, and they will continue to happen. The best way to minimize their impact is to catch them early by implementing, and more importantly executing, an effective pre-release testing plan.

BEST PRACTICES FOR AGILE TESTING

Now that you understand how agile testing should look, and why it is critical to your testing effort, here are our recommended best practices for getting a plan in place:

PARTNER WITH THE BUSINESS SIDE

Developers' work is not performed in a vacuum. The type of customer experience a digital property provides is often not created by the developer or the tester, but rather someone on the less-technical side of the business.

A team consisting of a developer, tester, and a business analyst should be ushering the project from beginning to completion, ensuring both business and quality goals are met. The business representative can also help push non-functional testing goals, such as user experience and security, along the way.

FIND THE RIGHT BALANCE BETWEEN AUTOMATED AND MANUAL TESTING

The benefits of automated testing are clear to any development team: faster execution, more timely feedback for developers, and more frequent test cycles. All of these lead to more rapid releases.

However, in the early lifecycle of a software product developed via an agile methodology, it may be more costly and take more effort to automate testing for that code base which is in rapid evolution. At this phase, manual testing is likely a more economic and effective approach to take.

Even when automation is established, it should be used in conjunction with manual exploratory testing to identify the types of issues that arise when real people use the product in unexpected ways. Automation can give a firm base of quality while manual testing by agile testers can polish the project and keep it on track.

EXPAND YOUR CAPABILITIES BY CROWDTESTING



The real focus for us in the way we work with Applause is to be able to get that rapid cadence and to be able to respond to market needs and get new features out to market quickly.”

ANDY NICHOL, SVP ENGINEERING, DOW JONES

Dow Jones ensures the Wall Street Journal, and all of its other media brands, consistently meet its readers' expectations by working feedback into its agile cadence. Learn more [here](#).

At the end of the day, no matter how solid of a plan you have in place, there is still a limited amount of time and resources to get everything done during every sprint. Tapping into the power of crowdtesting can enable you to easily overcome the most common barriers to agile testing, and accomplish more in less time.

Crowdtesting companies vet and maintain communities of qualified software testing professionals that can test software on their own devices, in their own homes, and extend QA into real-world conditions.

There are two primary types of feedback that you can get through crowdtesting. First, functionality feedback, which helps you decide if your digital property is working as you've designed it to. Then, there is also usability feedback, which enables you to confirm that the way you've designed it to work is aligned with the way your target audience expects it to work. Both are crucial to consult before a release, and [can be easily baked into your exist software development lifecycle](#) by crowdtesting.

The geographic reach of crowdtesting companies enables them to operate on a “follow-the-sun” business model. This means you can pass a build to a team of crowdtesters at the end of your day, and have regression results waiting for you when you get to your desk the next morning.

Crowdtesting is a reliable professional way to scale up and down to match testing to whatever cadence your agile development requires.

CONCLUSION

Agile provides an organized and adaptive approach to shipping out products, but brings its own set of unique challenges. Testing has proven to be one of the largest pitfalls for teams trying to implement agile.

While it is challenging to ensure testing keeps pace with development, you are not really agile if you are not agile testing. There are a variety of ways to overcome the challenges to accomplishing this approach.

Partnering with the business side, finding the right balance between automation and manual testing, and extending your internal capabilities with crowdtesting are examples of the low-hanging fruit that is out there right now. Tapping into strategies like these will enable you to accomplish a more proactive and thorough software development process, and ultimately, build a longer list of satisfied customers.

ABOUT APPLAUSE

Applause empowers leading brands to deliver flawless digital experiences for their customers on any device, in every location. The company's **market insights, user feedback and digital testing solutions** enable businesses to delight customers, drive topline revenue and innovate faster.

Applause ensures digital experience quality for websites, mobile apps, IoT products and in-store interactions in a way no other approach can – through its technology platform and managed **global community of over 300,000 professional and on-demand testers**. Only real people on real devices in real locations can provide the real issues and feedback that brands need to deliver great digital customer experiences. You can't hire, outsource or automate the increasingly converged digital-physical experience that defines real customer interaction, but you can replicate it with the crowdsourced approach provided by Applause.

Thousands of companies of every size – including Google, FOX, Nike, BMW, PayPal and Slack – rely on Applause to dramatically decrease both the time and the costs of delivering great digital experiences for their customers.

Learn more at www.applause.com

AMERICAS INQUIRIES

Applause U.S. HQ
100 Pennsylvania Ave -
Suite 500 Framingham, MA 01701
1.844.300.2777

EUROPE INQUIRIES

Applause Europe HQ
Obentrautstr. 72
10963 Berlin, Germany
+49.30.57700400

ISRAEL INQUIRIES

11 Galgaley Haplada
1st Entrance - 2nd Floor
Herzliya, Israel
+972.74.701.4240