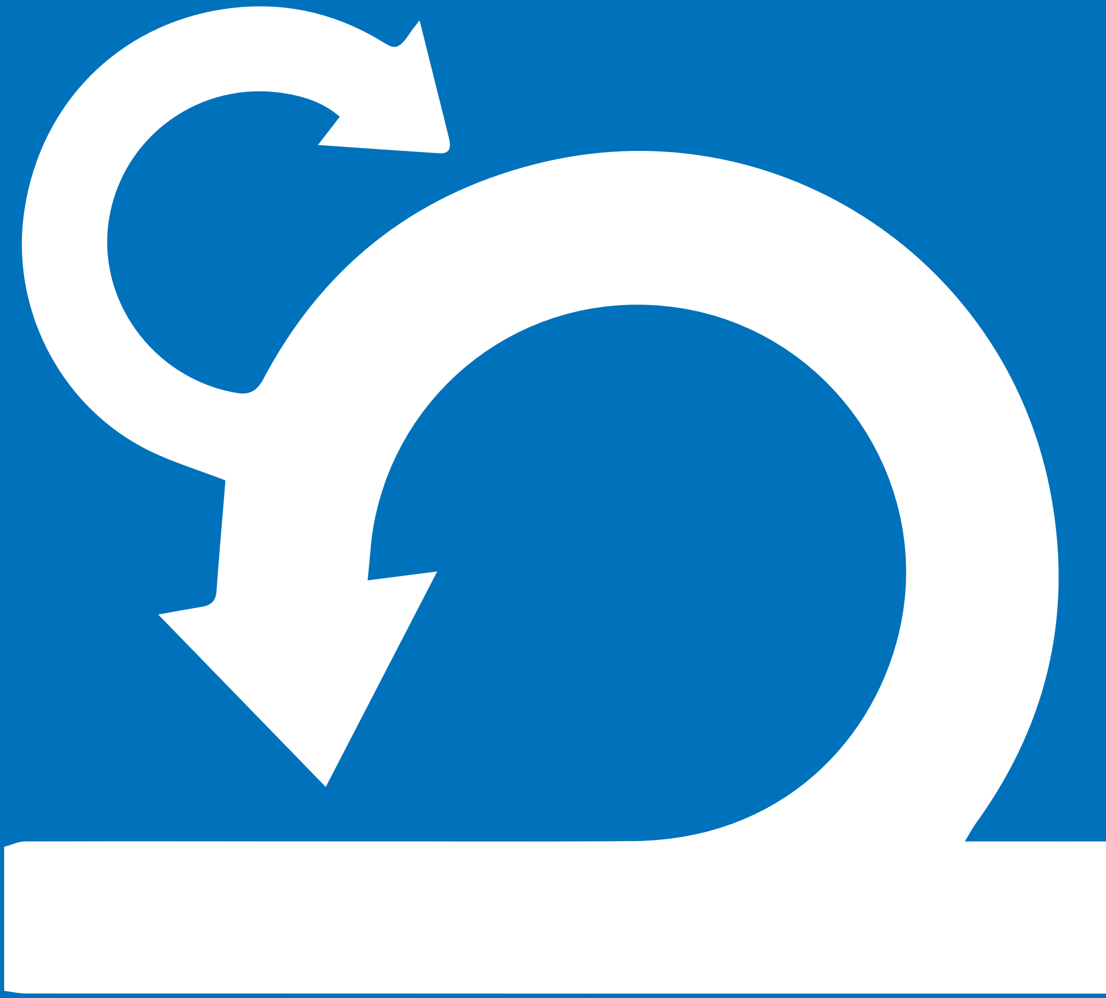
TESTING IN THE AGILE ERA



BASED ON ACTUAL USER EXPERIENCES & OPINIONS

IT Central Station
Unbiased reviews from the tech community

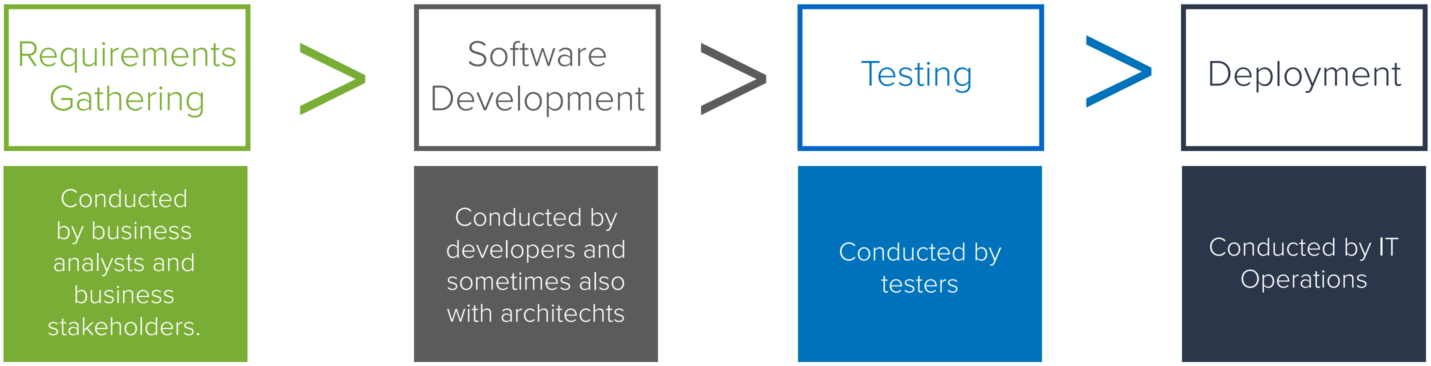
ABSTRACT

Agile development methodologies, as realized in DevOps and Continuous integration (CI) of software code, threaten to make a mockery of traditional testing processes. How can you test a code base that's not only live in production but being changed on a daily or even hourly basis? Solving the challenge involves rethinking testing practices. It also requires new tools. In this paper, we look at recommendations for testing tools and practices from members of IT Central Station who are tasked with testing in a CI environment.

CONTENTS

INTRODUCTION

Agile software development methodologies, which have been on the rise in IT for the last 15 years, challenge the traditional "waterfall" approach to development. Rather than going through the process of gathering requirements, developing a complete code set, followed by testing, the agile approach works in rapid cycles focused on the most pressing requirements at a given point in time.



Requirements Gathering > Software Development > Testing > Deployment

Conducted by business analysts and business stakeholders.

Conducted by developers and sometimes also with architechts

Conducted by testers

Conducted by IT Operations

Agile is potentially transformative, given its ability to release software much more quickly than the old approach. Agile is in ascendancy in IT. Industry surveys vary in their estimates, but research suggests that about 90% of development organizations are either completely or partially agile.

While it offers many strategic and operational advantages, agile methodologies threaten disrupt traditional testing processes. The pace of development is so fast it outstrips standard testing practices. Then, with implementations like DevOps and Continuous Integration (CI), it can be almost impossible to keep up. Indeed, how can testers assess a code base that's not only live in production but being changed on a daily or even hourly basis?

Solving the challenge involves rethinking testing practices. It also requires new tools. In this paper, we look at recommendations for testing tools and practices from members of IT Central Station who are tasked with testing in an agile environment.

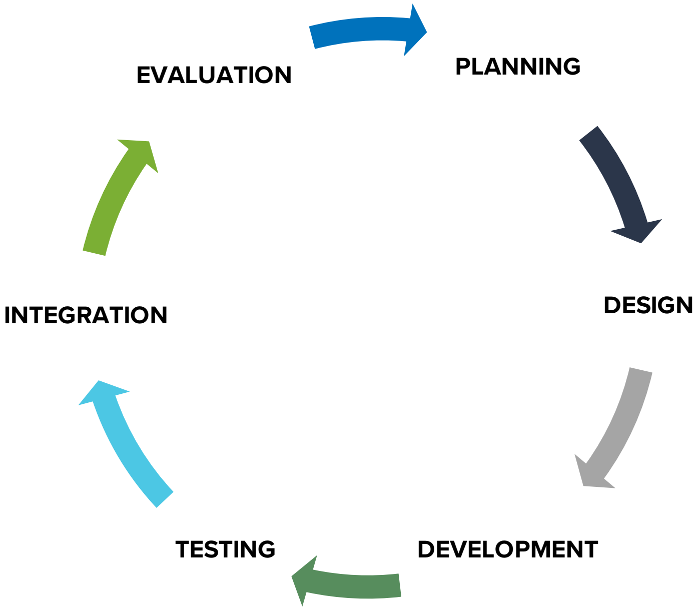
Agile Methodologies and Their Impact on Testing

Agile software development has spawned a variety of operational processes. These include “Scrum,” an iterative, flexible process. The scrum, a term borrowed from the sport of Rugby, involves a group of people who agree to implement all agreed-upon features in a set period time, e.g. 30 days. There’s also “Feature Driven Development” (FDD), which uses short iterations, e.g. two weeks, focused on the achievement on the build of a specific feature. Extreme Programming (XP) is a flexible and lightweight method, often involving collocated teams. These are just a sampling of agile methodologies in use today.

Agile methodologies can be operationalized in a number of different ways. Two approaches that often go together are DevOps and Continuous Integration (CI). DevOps refers to the merging of previously independent software development (the “Dev” in DevOps) and testing and IT operations teams (The “Ops). In the waterfall days, developers would finish their build and “throw it over the wall” to testing and IT operations for deployment. With DevOps, development and operational activities are performed collaboratively between teams or even through unified teams. DevOps is a good operational fit for agile methodologies. It helps speed deployment of code that’s being rapidly generated.

CI is the practice of adding new code continuously to an application that is deployed in production. It takes special tools, as not all apps are able to absorb new code without being uninstalled and reinstalled. CI is great for applications that must adapt quickly, such as consumer-facing web applications that respond to customer feedback in fast cycles.

CI, which may take place with or without DevOps, is an effective complement to agile methodologies. It makes it possible for agile teams to create small blocks of code that can be put into product almost instantly. Some CI shops update code every day or even more frequently. In contrast, under waterfall methods, an application might get updated twice a year. Figure 3 shows the traditional software releasing process.



Agile methodologies can be operationalized in a number of different ways. Two approaches that often go together are DevOps and Continuous Integration (CI). DevOps refers to the merging of previously independent software development (the “Dev” in DevOps) and testing and IT operations teams (The “Ops). In the waterfall days, developers would finish their build and “throw it over the wall” to testing and IT operations for deployment. With DevOps, development and operational activities are performed collaboratively between teams or even through unified teams. DevOps is a good operational fit for agile methodologies. It helps speed deployment of code that’s being rapidly generated.

CI is the practice of adding new code continuously to an application that is deployed in production. It takes special tools, as not all apps are able to absorb new code without being uninstalled and reinstalled. CI is great for applications that must adapt quickly, such as consumer-facing web applications that respond to customer feedback in fast cycles.

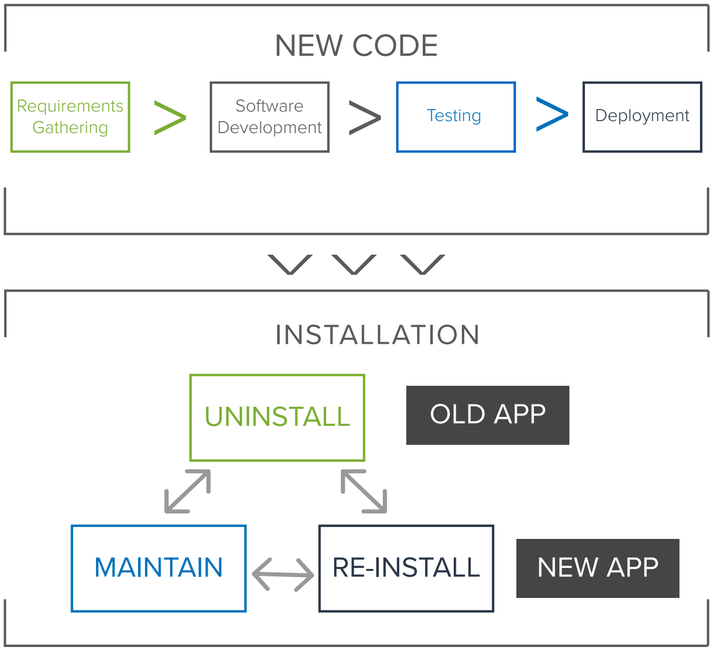
CI, which may take place with or without DevOps, is an effective complement to agile methodologies. It makes it possible for agile teams to create small blocks of code that can be put into product almost instantly. Some CI shops update code every day or even more frequently. In contrast, under waterfall methods, an application might get updated twice a year. Figure 3 shows the traditional software releasing process.

All of this is great unless you're a tester. Agile methodologies and their various operational implementations wreak havoc on testing. In an earlier era, in "olden days" of, say, 2010, testers had time to plan and execute tests on a full application. The team had a pretty good idea when they were going to get the code for testing. They knew how long they had to conduct tests. They had a sense of what parameters they would need for testing. This is no longer reality.

Today, software testing teams are thrown piece of applications for rapid testing. Some of the most basic fundamentals of traditional testing no longer apply. With CI, for example, it's nearly impossible to test how a build is going to behave in production. For some organizations, deployment is testing.

One additional wrinkle in all this is the emergence of new software architectures that further compound testing processes. The rise of RESTful APIs, containerization and microservices, to name a few examples, introduce untold numbers of unknown third parties into production software. It's hard enough to test for known factors in an application. What happens when, all of a sudden, the tester must consider the effects of cloud-based software in outside entities interacting with the application in unpredictable patterns of traffic and load?

WATERFALL DEVELOPMENT & SILOED DEPLOYMENT



Rising to the Agile Testing Challenge

It's not quite fair to say, "This isn't your father's software testing environment." If you've been involved in testing for more than 5 or 10 years, it's not even your own testing anymore. Testing has been forever changed by agile methodologies and related trends in IT. Many IT organizations have responded by revamping both their testing tools and practices.

Adopting new tools seems to be paying off. Reviewers on IT Central Station make this point consistently. For example, a Consultant at a tech services company noted that with his testing tool, “You could [We] cut the technical part of the test automation processes by about 50%.”

> “You could [We] cut the technical part of the test automation processes by about 50%.”

What are expert users looking for a in a new testing tool? A Director of Testing & Quality Assurance at a software development firm explained what he wanted in a testing tool for agile methodologies. He valued a testing tool with “support for containerization and continuous learning, adapting to our needs like support for DevOps practices.” He added that these capabilities are “paramount in our work.” In this spirit, the following guidance, shared by professional testers on IT Central Station, offers recommendations on best practices and tool selection for this new era of software testing.

BRINGING DEVELOPERS INTO TESTING

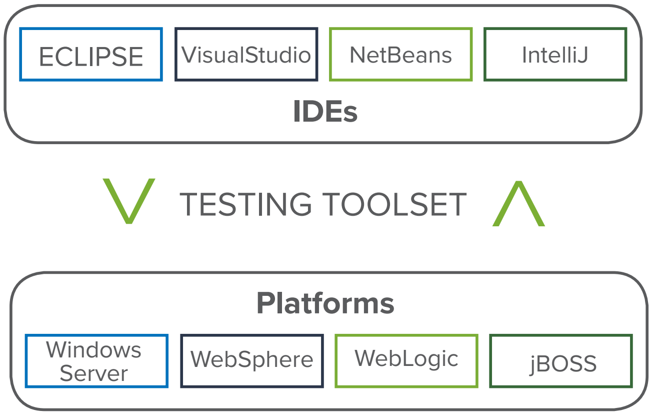
One solution for the challenges of testing in agile development is to bring the developers into the testing process itself. A Technologies Consultant at a tech company with 1,000+ employees explained, “The most important thing about [our testing solution] is that it gives us the opportunity to introduce developer testing. They can use the same language that they’re using for their unit testing, so they can contribute to that in the very beginning of the lifecycle instead of after the application has been fully developed, at which point it is tested and comments are made on the features. We can put the developer into the testing scenario.”

It helps to have a testing tool that uses a developer-friendly language. As a Test Automation Architect at a large healthcare company explained, “What’s awesome about it is you can use the same language the developers already use. [Our testing tool] integrates with our developers work-flow. Our developers can now perform automation using familiar tools, programming language, and IDE. This helps get the whole team involved in test automation.” He then added, “Testers should be using the same tools as the developers, making the development process easier.”

Usability is also a factor in bringing developers into testing. A Director of Service Transition and Quality Management at a health insurer said that his testing tool is, “used a lot more by our less experienced testers and developers who want to get more point-and-click type automation...We pretty much use it to streamline workflow and enable productivity in a business context.”

FLEXIBILITY IN TOOL SETUP

Testers on IT Central Station consider flexibility in tool setup to be an important criterion for testing solutions in this agile age. An Enterprise DevOps Leader, Program Manager at a media company with 1,000+ employees like the fact that his testing solution has “flexibility across different platforms.” He noted, “For example, I can use it whether it’s Eclipse using any of the programming languages. The multi-platform usage of [the tool] is really useful.” The Technologies Consultant at the tech company was switching to a new testing tool because, “It will give us the most flexibility with more ways of working with the CPU.”



INTEGRATION WITH IDES AND CI TOOLS

Given the rapid-fire nature of application development today, it makes sense that testers favor testing solutions that work well with Integrated Development Environments (IDEs) and CI tools. As the Test Automation Architect at the healthcare company explained, his testing tool "integrates with our developers work-flow. Our developers can now perform automation using familiar tools, programming language, and IDE. This helps get the whole team involved in test automation."

A Software Quality Assurance Supervisor at a Consumer Goods company echoed this sentiment, saying, "Integration with other IDs [is valuable to his organization] It being in the IDs that the developers are using is big." He added, "With my staff, it's retention. Keeping them happy, things like that. That's important. Happy folks are more productive, but also as we spin up agile delivery teams having that integration where the developers are more or less the kings, we can sell that, and say 'Hey, use this product.' It's already sitting there in Eclipse, sitting there in Visual Studio, so that's pretty big."

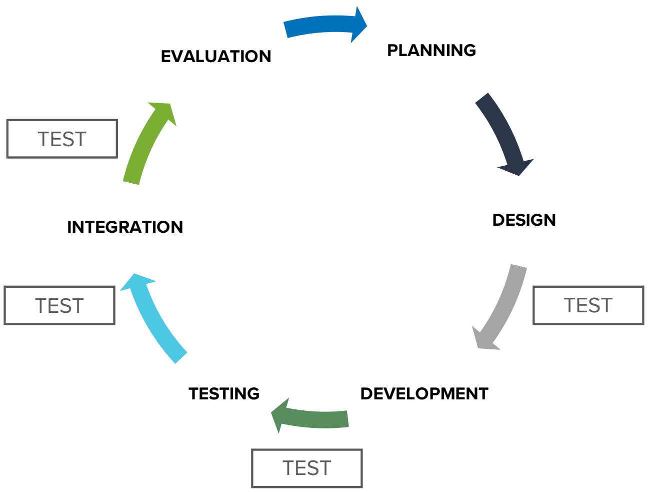
It helps if testing integration continues from development to production, especially in a CI environment. A Senior Automation Engineer at an insurance company with 1,000+ employees made this point, arguing that he valued his testing tool's "ability to implement with Jenkins for DevOps." Jenkins is an open source automation server that is very popular in DevOps and CI.

THE ABILITY TO HANDLE COMPLEX ALGORITHMS

Testing has come to rely more on complex algorithms as development cycle has sped up and become more uncertain. For this reason, testers are looking for tools that can handle complex algorithms. As a Director of Testing & Quality Assurance at a software development firm explained, "I work a lot with guys who work with meta-mathematics, like applied maths and quantum mathematics. So for us, the most important feature is the ability to handle complex algorithms, such as fuzzy logic techniques which is the first step towards artificial intelligence in our field."

END-TO-END TESTING CAPABILITIES

Continuous Integration demands end-to-end testing. The Director Testing & Quality Assurance at the software development firm commented on this issue, saying he wants a tool that offers "continuous assessment, continuous insight and quality, as well as testing that continues to be driven onwards." He further said, "We have to think about the end-to-end stack, from problem definition to solution delivery,



a solution that sees the whole end-to-end lifecycle of the application. The whole vision is important for me."

CUSTOMIZATION

Agile development requires the ability to customize testing solutions. For example, the Technologies Consultant at the tech company described his customization needs by saying, "With [our testing tool] we are trying to create a kind of hybrid infrastructure where we can use our existing scripting and in the same infrastructure we can use [the testing tool]."According to the Senior Automation Engineer at the insurance company, the most valuable features of his testing tool are, "the ability to customize, if needed, and to integrate with ALM [Application Lifecycle Management] because it's our test management."

AUTOMATION

Faster development cycles favor automated testing. The less manual testing work, the more testing can keep up with the fast-pace of DevOps and CI. As the Software Quality Assurance Supervisor at the Consumer Goods company put it, "Obviously, automation is what we need to do to get more testing done in a shorter amount of time. With more people doing the automation, we are able to get more tests automated, more tests done, so overall we are turning over a more quality product."

> "Our customers demand these results quickly, and this solution was able to deliver extremely well."

The Director Testing & Quality Assurance at the software development firm commented, "Within a week, two of my global customers were able to leverage their automation through this solution. The adaptability of how this slotted in was just amazing, which was incredibly efficient. Our customers demand these results quickly, and this solution was able to deliver extremely well."

Automation can mean better usability, with team productivity benefits. According to the Director, Service Transition and Quality Management at the health insurer, "[Our solution] is used a lot more by our less experienced testers and developers who want to get more point-and-click type automation. Then obviously I also manage the automation team, so we use [the tool] for a ton of all of our XLC automation as well as anything with a UI. We pretty much use that to streamline workflow and enable productivity in a business context in our business unit as well as in our IT shop, so just reducing workload on IT people as well as testing. It's used pretty extensively beyond that."

EASE OF SETUP

Things are complicated enough in the agile world that

testers want tools that are easy to set up. A Software Quality Assurance Supervisor at a consumer goods company was pleased with his tool. He said the setup was, "Fairly straightforward. For us, it's a simple download, pointed at our concurrent license server and we are good to go. I would characterize it as easy." A Software QA Lead at a healthcare company concurred. The Director, Service Transition and Quality Management at the health insurer explained that while he has a tools team reporting to him, "I did get involved in the planning and the strategy of how we're going to do it. My team said that first installation is relatively easy."

SCALABILITY

Testing tools need to scale. The Director Testing & Quality Assurance at the software development firm praised his tool's scalability. He said, "It's designed for scalability. Cloud maps and complex ecosystems, which we've got today, even the internet, they're all made up of nodes. And being able to scale is paramount to evolving those nodes. This solution gives us the ability to scale however we like, which is why we use it. There's no use having 50,000 nodes that run wild and can't be controlled, so allowing us to control it is the value of the product."

A Solution Architect at a tech company find his tool to be "extremely scalable as far as the testing that we've done with our customers. They've all been really satisfied with the scalability." A Director, Service Transition and Quality Management at a health insurer simply commented on his tool, "We're already at enterprise scale, so it's used across the enterprise." The Enterprise DevOps Leader, Program Manager at the media company praised his testing solution remarking, "It's very scalable. Given the power that it provides in terms of any programming language and any development platform you can use it on, such as Eclipse or any ID platform."

CONCLUSION

As waterfall software development gives way to agile methodologies and its various organizational and architectural manifestations, testing teams are looking to new toolset to keep up with the faster pace and greater variety in testing scenarios. Testers want tools that help them deal effectively with DevOps and Continuous Integration (CI) as well as architectures like containerization.

According to expert users writing on IT Central Station, some of the most important selection criteria include qualities like flexibility of setup and the ability to integrate with IDEs. Testers also want end-to-end testing abilities and a high degree of automation. They want compatibility with complex algorithms. New testing tools for the agile age must be efficient and effective, enabling testing teams to play their critical role in the application development process no matter how it changes over time.

ABOUT IT CENTRAL STATION

User reviews, candid discussions, and more for enterprise technology professionals.

The Internet has completely changed the way we make buying decisions. We now use ratings and review sites to see what other real users think before we buy electronics, book a hotel, visit a doctor or choose a restaurant. But in the world of enterprise technology, most of the information online and in your inbox comes from vendors but what you really want is objective information from other users. IT Central Station provides technology professionals with a community platform to share information about enterprise solutions.

IT Central Station is committed to offering user-contributed information that is valuable, objective and relevant. We validate all reviewers with a triple authentication process, and protect your privacy by providing an environment where you can post anonymously and freely express your views. As a result, the community becomes a valuable resource, ensuring you get access to the right information and connect to the right people, whenever you need it.

www.itcentralstation.com