

**Borland**<sup>®</sup>

THE OPEN ALM COMPANY

## Testing Tools to Support Agile Software Delivery

---

The Critical Role of Automated Functional  
Testing in Enterprise Environments

WHITE PAPER  
SEPTEMBER 2008

## Contents

Executive summary .....	3
Automation and Agile Software Delivery .....	4
The Role of Automated Testing in Agile Development .....	4
The Need for Speed: Accelerating the Code-and-Test Process with Fast, Automated Test Scripts .....	4
The Need for Repeatability: Performing the Same Tests and Scripting Tests Against the Right Acceptance Criteria .....	5
Borland Case Study: Using Automated Testing to Support Agile Delivery .....	5
The Starting Point .....	6
Integrating Testing Teams to Create Complete Engineering Teams .....	6
Choosing the Right Test Automation Tools .....	7
Borland SilkTest for Test Creation .....	8
Borland SilkCentral Test Manager for Test Management and Synchronization .....	8
Java as a Scripting Language .....	8

## Executive Summary

More and more companies are moving to agile software delivery approaches. But agile delivery brings with it a new set of challenges; among them, functional test automation - opinion is divided in the Agile community on the value of automated testing. However the reality is that software teams must manage quality if they are to avoid operational risk. Businesses are encouraged to adopt an agile process that fits their unique needs. Nothing about agile is pre-determined or dictated so it's no surprise that businesses find it challenging to implement agile as a trusted business process – using tools to automate their agile delivery process is yet another variable that complicates the decisions they must make.

This paper discusses the benefits and offers some guidance on incorporating automated functional testing and testing tools into agile software delivery environments. You will learn about the critical role that automated testing plays in helping companies implement and support an Agile process. The paper discusses aspects of Borland's agile transformation as a reference point or use case. You'll see how a geographically distributed team overcame challenges and leveraged automated testing and other tools to ensure a successful transition. We'll discuss the lessons learned in our transformation and as a result how Borland's automated tools are being developed to support and enhance Agile delivery methods.

## Automation and Agile Software Delivery

In the world of agile software delivery, requirements are raised, changed, and reprioritized in much shorter timescales. At the same time, the functionality that you deliver must be fully tested to ensure that it meets the needs of end users and the business. As software volume and complexity increases, it's clear that manual processes will fail to deliver fast and repeatable results that underpin the Agile proposition. Inevitably without automation, quality will suffer as testing is marginalized to achieve on-time delivery.

Regardless of whether your development team is considering going agile, making the transition, or already operating in an agile mode, it's critical to consider how technology can support this new way of working – particularly in the area of testing. According to Forrester; “there's room for improvement in agile processes – for example, through better support for agile from mainstream tools for test automation, test management, and requirements management. Forrester notes that in an agile process, testing is continuous and imperative, as features are not “done” until all associated test cases pass. Moreover, “Testing has to be as automated (and as reasonable) as possible because of regression burden.”

The issues facing organizations implementing Agile testing automation are;

- Operating Agile and traditional testing methods side by side
- Identifying tools that are usable and flexible
- Allowing non-technical and non-testing specific members to actively contribute
- Enabling open-source integration
- Enabling test-driven, or model-driven, development

## The Role of Automated Testing in Agile Development

Let's qualify why testing needs to be automated in an agile delivery environment. The people involved in testing are part of the delivery team – not an isolated group that developers hand code off to as a final step to release. Ideally, they sit side by side with the developers, who, as they create code, pass it to testers early in the process for evaluation against acceptance criteria. Since capability is built iteratively and the team needs to maintain velocity – the code assets need to be verified quickly. For agile to truly succeed, functional testing has to be quick, iterative, and responsive.

Specifically, automated functional testing offers a number of efficiencies that cannot be achieved with manual testing.

For example, it can:

- Greatly accelerate the code-and-test process by supporting fast, automated test scripts
- Ensure the repeatability of tests, to ensure regression testing from sprint-to-sprint, iteration-to-iteration.
- Enhance test efficiency further via robust, yet flexible, test management processes
- Avoid the inherent inaccuracies that manual processes inject in to the process – particularly when time is tight.

### **The Need for Speed: Accelerating the Code-and-Test Process with Fast, Automated Test Scripts**

Automation enables testers to create simple, reusable scripts, which they can deploy to save time and increase the consistency of testing across similar user stories, story points, or requirements within and across projects. They can be run rapidly and iteratively, developed from the User Story to drive the functional capabilities, enabling Test Driven Development. It significantly lightens the workload of testers and eliminates the need for late night and weekend testing marathons that can burn teams out.

---

<sup>1</sup>Teleconference: “Implementing Agile in a Waterfall Environment,” Slides 26 and 33. Carey Schwaber, Senior Analyst. Forrester Research, March 17, 2008.

## **The Need for Repeatability: Performing the Same Tests and Scripting Tests Against the Right Acceptance Criteria**

Regression testing requires that 1) you perform the same tests each time you test a particular piece of code and 2) that the test is scripted against the acceptance criteria of each respective user story. Whenever code changes, (or is extended to include new capability) you need to rerun all functional tests for all user stories up to the latest change to ensure other user stories weren't impacted inadvertently. (With agile development, regression testing should typically be performed at the end of every iteration - in some cases this means daily.)

Repeatability is nearly impossible to achieve with manual testing due to human error, variability, and inconsistency. People simply can't remember exactly which tests they performed for each piece of code for the last iterative cycle – and even one oversight can cause problems in final code. But with automated, repeatable functional testing and regression testing, you can execute tests consistently whenever you need to.

## **Borland Case Study: Using Automated Testing to Support Agile Delivery**

Borland has first-hand experience with an enterprise Agile transformation, we have observed the critical role that automated testing and centralized test management can play in enabling a successful transition. In 2006, we started our transition to Agile with 350 engineers working on a broad portfolio of development projects from 13 different locations worldwide – to an agile delivery approach.

We took an iterative approach in the company's agile journey, transitioning by geographic region rather than all at once. Approximately 60%+ of Borland teams now use formal agile methodologies – all supported by Borland's ALM products and the benefits have been tremendous. However we respect the observation made by Forrester that agile adoption can be hard for organizations because of changes that need to be made in the following areas<sup>2</sup>:

- Organizational structure
- Roles and responsibilities
- Requirements management
- Testing
- Project and portfolio management

According to Forrester, most enterprises are not set up to accommodate the kind of continuous and imperative testing required to support agile development. Moreover, Forrester notes that, "In an enterprise context, some exceptions usually have to be made for especially resource-intensive testing, <such as> performance, security, usability, UAT, and release testing."<sup>3</sup> [GG3]

Borland serves as a useful case study for others that are trying to determine the best testing strategy to adopt within their Agile practice. Within our own organization team members are geographically separated, but as we've proven, this does not preclude a team from implementing agile methods. We were able to implement a sensible balance of tooling and automation that helped our Agile teams work effectively.

## The Starting Point

By way of example; our development facility in Linz, Austria had three development teams using a traditional waterfall approach to develop Borland's Silk testing solutions. After some analysis we decided that within our transformation to Agile across multiple teams, tool automation was going to be a critical component.

Obviously the transformation was not predicated on simply being able to use tools – so as with many other teams, the first step was to define new roles such the Product Owner (PO). The PO's main responsibility is to facilitate communication between business and technical stakeholders and then communicate their software requirements and changes to the development and QA teams. This was supported through daily stand-ups using tools to enable communication and collaboration for ALL of the team members irrespective of their actual location.

The Linz team transitioned to four-week iterations, allowing development teams to focus on smaller units of work – a step toward becoming more agile. The team maintained User Stories in Borland CaliberRM and validated these with QA for testability. The User Stories were linked to Borland SilkCentral Test Manager, enabling QA to develop the test cases and their associated automated functional test scripts in Borland SilkTest. The result was an improved, collaborative process that enabled the creation of high-quality, testable requirements (expressed as user Stories). Now developers and testers had clearer requirements statements and could implement testing earlier in the development process, effectively enabling test driven development and maintaining traceability between User Stories, code, tests and results.

## Integrating Testing Teams to Create Complete Engineering Teams

Following the successful implementation of a testing process and infrastructure that improved our Agile effort, we then moved the QA team into the product development teams, members of QA were allowed to self-organize into product-specific teams, taking the relevant automated functional test scripts with them. Merging the teams resulted in :

- Improved communication because the team could develop the functional test scripts in parallel with the code based on the same requirements and changes.
- Improved ability to meet quality objectives as everyone was now proactively responsible for quality
- Improved test coverage via test pairing and by sharing libraries of unit tests (SilkTest scripts created by developers) and functional tests (SilkTest scripts created by testers) to prevent duplications. This reduced the effort required to develop automated scripts and guaranteed code-and-test coverage.
- Earlier isolation of defects and quality issues leading to improved velocity.

To maintain quality standards and help teams build testing competency we created another new role called “QA coach” – a single resource who provides QA expertise for all three teams. Under the direction of the QA coach, (and the use of SCTM to organize and plan scripts, and StarTeam to store and version control them), people could perform their roles more effectively, share and re-use Borland SilkTest scripts across the teams.

## Choosing the Right Test Automation Tools

We've proven that Agile scales effectively when it's supported by the right mix of people, process and tools. While many may argue that automation has no place in Agile, the realities of geographic separation, multiple teams and limited resource dictate a more pragmatic approach.

Automated functional testing and centralized test management play a vital role in helping development teams implement an Agile delivery process. But the majority of businesses today continue to use manual processes, which can make it difficult, if not impossible, to implement and scale agile methods within an enterprise. Manual testing slows down production, and is error prone – it does not support re-use and provides no predictable repeatability.

On the flip side, those companies that have deployed automated testing tools in traditional projects run into other challenges with regard to agile development. For example, most software testing products force companies to operate in a certain way (for example, using a waterfall method) rather than an agile, company-specific process.

In contrast, because of our direct experience with an Agile transformation we've ensured that Borland Silk products can help you implement, and enhance, an Agile delivery process while ensuring quality, reducing risk, and lowering costs. Employed by thousands of enterprises worldwide, Borland's testing tools can support any development process – agile or traditional – and be configured to support any development team's preferred processes. These tools also enable teams to implement automated testing that is rapid, repeatable, and accurate. And because they are part of an integrated Open Application Lifecycle Management suite supporting requirements-based testing, lifecycle quality management, development teams do not operate in isolation. Everyone across the enterprise has visibility into the agile delivery process and can participate in it as appropriate.

As an example, SilkTest is commonly used to record and replay against a GUI-front-end in an application and the record and playback capabilities help to enhance the speed with which tests can then be generated and run in any traditional test environment. There also exists a powerful ability to reference portions of the application where there may yet be no UI; a very common consideration in more Agile development projects. One of the biggest challenges facing Agile development projects is that as the application rapidly develops, the use of any test automation rapidly changes too. Scripts that deal with the UI in these environments are very quickly ineffective as objects and even structure on the front end change.

SilkTest assists these project teams in a couple of very similar ways: First, the ability for SilkTest to directly call application DLLs within its script by-passing the UI. This allows teams to secure automation points onto more concrete areas of the application. The second way is through SilkTest's Silk4J environment, whereby automation efforts can also connect to lower-level Java portions of the architecture.

In either instance, SilkTest assists the teams, by providing an ability to work with much more stable levels of the application architecture, not just the UI. These tests can also create placeholder artifacts within the respective scripts to reference items that have yet to be built. Once those items are built, the automation logic can replace the stubs, creating fully functional automation scripts.

### **Borland SilkTest for Test Creation**

Borland® SilkTest® provides powerful functional and regression test automation. Intuitive GUI record-and-play capabilities, coupled with stable, easy-to-use testing languages, enable teams to build functional and regression tests that do not break with minor changes to the application. The software also makes it easy for developers to perform frequent, regular, and automated regression and functional software testing.

### **Borland SilkCentral Test Manager for Test Management and Synchronization**

Borland® SilkCentral® Test Manager™ is a powerful, all-inclusive software test management tool that builds quality and productivity into the testing process. Borland SilkCentral Test Manager integrates with other Borland tools and technologies, as well as third-party application lifecycle management tools. This integration ensures that software testing becomes a managed process that spans the entire software development life cycle and helps to align business and development priorities.

SilkCentral Test Manager also streamlines build verification tests that are critical to supporting agile development, as each team needs to ensure that the new code they build each day doesn't cause unexpected problems in other areas of the application. These tests can generally be performed in two hours on each build and across different configurations, enabling each scrum team to deliver working software at the end of each day.

## **Java as a Scripting Language**

Borland continues to develop innovative approaches to support Agile delivery such as the addition of Java as a Scripting Language for SilkTest\*. This capability allows developers and testers to create scripts for SilkTest in a familiar development language – Java. This is facilitated through an Eclipse plug-in to allow scripts to be generated, in the same language as the development itself. Scripts are created and run by the Open Agent capabilities of SilkTest.

### **SilkTest and FitNesse**

FitNesse provides a keyword driving, Open-source framework that allows testers to specify tests at a business level (in a table), where actions are linked with expected results. FitNesse then references selected actions and results against automated tasks in SilkTest. Through the Java as a Scripting Language ability, we create libraries that reference objects and functionality with room for parameters as defined in the test. Similarly we craft verification routines as shared classes that can also be used. Further to that, using the Open Agent, we can call these scripts by way of FitNesse's external tool execution functionality. What this ultimately gives us is the ability for non-technical users to design test scenarios in the web browser/wiki environment (FitNesse), which will then use SilkTest without any additional scripting to drive and pilot the application under test. Results are gathered from the SilkTest .RES files, and pulled into the FitNesse browser.

\*Borland Silk 4Test, will continue to be supported and developed.

## **For More Information**

To learn more about Borland Silk and how Borland can help you with your Agile transformation, please visit [www.borland.com](http://www.borland.com)

---

Borland is the leading vendor of Open Application Lifecycle Management (ALM) solutions - open to customers' processes, tools and platforms – providing the flexibility to manage, measure and improve the software delivery process.