

# Agile Testing Best Practices

## Introduction

The testing phase of software development sometimes gets the short shrift from developers and IT managers. Yet testing is the only way to determine whether an application will function properly at deployment. Without an effective testing strategy, companies sometimes blindly take on significant risks that go well beyond simply having poor functioning software. Organizational ramifications can be fierce, including the risk of:

- Customer/end user alienation, or brand injury,
- Competitive threat, and even
- Product revenue loss.

At the development level, an oversight in testing can trigger countless technical and project outcome challenges:

- Protracted development cycles, or the reverse, a shortened testing cycle to compensate for longer development phases,
- Missed use cases triggered by developer-tester misalignment or poor communication and resulting in compromised quality,
- Cost overruns or simply poor resource estimations.

Agile testing mitigates these risks and presents an effective solution to ensure that you achieve the technical, project and business goals of the development process. In this whitepaper, we discuss Agile Testing, why it's important, and *how* to implement it.

### What is Agile Testing?

Agile Testing is a software testing practice that follows the principles of Agile software development. Agile development integrates testing into the development process, versus having it as a separate phase. Testing therefore is an integral part of the core software development and actively participates throughout the software coding process. Agile Testing involves a cross-functional Agile team actively relying on the special expertise contributed by Testers. This allows the combined team to better meet the project's defined business, software usability, quality, and timeline objectives. Agile teams use a "whole-team" approach to "bake in quality" to the software product. This approach allows the team to work at a sustainable pace because testing occurs in real time, allowing testers to collaborate actively with the development team and giving them an

ability to identify any issues and transfer those into executable specifications that guide coding. Testing and coding are done incrementally and iteratively, building up each feature until it provides enough value to release to production.

### **Good Agile Testing Practices and Traits**

Effective agile projects generally address the following important elements:

- Includes Testers and QA Managers as full members of the Agile development team.
- Leverage Testers as active contributors in planning and requirements analysis.
- Instills team value whereby each team member is responsible for the results, including quality.
- Promote the importance of Testers and encourage continuous feedback sharing with the programmers and the customer.
- Testers actively participate in meetings to define the main business flows.
- Testers complete short iteration activities alongside developers.
- Encourage maintenance of traceability between the requirements, test cases and bugs.
- Testers contribute to user story improvements.
- Leverage the specialized skills of test-driven development, including unit testing, continuous integration and unit level.
- Leverage Automation testing as a key way to do regression testing.

**How does Agile Testing work?** The process diagram at the back of this paper describes how the process works? Please note that for the purposes of this paper, some steps of the Scrum have not been included in diagram.

1. The combined team, including testing, takes responsibility for analyzing the business requirements (e.g. user stories). They together define the sprint goal.
2. The QA team defines the testing scope (i.e. test plan). That is then validated by the whole team and the client.
3. Simultaneously, while the development team starts the implementation, the QA team begins work on the test case design. These are properly documented and handed over to the client and the development team for review. This is to ensure the complete test coverage avoids unnecessary or redundant test cases.
4. QA defines along with the development team and the client which main flows (test cases) will be automated.

5. When code is ready to test, QA agrees with development to do quick testing on development environment, in order to identify the early stage defects so developers can fix them during the next round, on priority basis, and then progressing with further development. This iteration continues until the end of the code implementation.

6. Automated test cases are run daily. Any defect found is reported and fixed, based on its priority.

7. The QA team then begins testing on the QA environment. Any defect at this stage is again reported. At the end of the iteration the team determines, along with the client, which defects are to be fixed in the current iteration.

### **Automated Testing-Regression**

Automation is a critical component of Agile testing. It would be impossible to keep pace with the Agile development schedule otherwise. Automation is used to run regression testing. The combined team (Developers, Product Owners and Testers) usually predetermine, at the start of the project, which parts of the software will be tested using automation.

Continuous integration/builds, unit, functional and integration test execution as well as continuous or automated deployment are common areas where automation may work better than traditional tests.

The entire project team agrees upfront on which of the main flows will be automated. They also determine at this point how to prioritize defects identified by automation, and how to fix that during sprints.

Automated tests consist of unit tests, capable of verifying even the most minute segment of software. Furthermore, it does so rapidly. This makes it possible to execute the test set multiple times per day, per hour or even more frequently if needed.

The benefits of automation include:

- Allows re-use of tests
- Enables faster execution for the most important test cases.
- Facilitates greater test coverage
- Delivers higher test accuracy and identifies defects sooner
- Facilitates regression testing

## **Advantages of Agile Testing**

- Testing requirements are discussed and refined as a team (stand-ups / Scrums) allowing combined team to better address the business/technical aspects of the requirement. This enables overall alignment and prevents misunderstandings.
- Agile process often requires having an entry and exit criteria for stories (a compression of things to do in a particular release/iteration). Agile testing ensures that each requirement is well defined and is measurable, allowing you to determine whether the requirement was actually completed or not.
- QA participates in the big picture requirements writing stage. This ensures testing estimates aren't overlooked.
- Automated tests are fully leveraged to implement Regression.
- Quality becomes the combined team's responsibility, rather than just solely that of QA. The entire team agrees to the testing strategies, test cases and defects prioritization plan.

## **Disadvantages of Agile Testing:**

If the ultimate goal or big picture requirements are unclear, the details can become muddled. Normally for new products the software architecture takes a path based on the initial requirements. If the requirements change (as allowed for in Agile), the following scenarios can occur:

- Engineering struggles to adapt to changes because significant effort has already gone into the initial requirements development and testing process.
- Challenge in estimations and sizing requirements given people dependency. Sometimes QA gets the short shrift since it's logically the last task in marking a user story done. Therefore, any delay in the prior development task risks impacting QA timelines. Sometimes, QA is prevented from executing a test case for the whole iteration, leaving QA to struggle to finish the task.
- Not asking the right questions. It very dangerous for QA not to ask questions, especially at the point where the use story is picked up for implementation. Ensure daily team meetings avoids this problem.
- Addition of new user stories into the current iteration. If QA is not included in the addition of the new use story, the appropriate commitments, estimations are not built in, resulting in misalignment and protracted timeframes.

## **Agile Testing Portfolio**

Testing tools include:

Selenium, IBM-Rational Functional Tester, MS-Team Foundation Server, HP Mercury, Apodora, Ranorex, Eclipse, Watir, Watin, Paros proxy, Nunit, MS SQL,

Jira, Rally, TestLink, Jmeter, Pywinauto, Virtual Machines, WinSCP, XML, XLST, etc.

Relevant Technologies include:

C#, Java, Python, Silverlight etc.

The key message is that testing should be considered a crucial process to the final software outcome. Employing an Agile Testing approach can ensure overall alignment between Testing and Software Development, yielding overall project outcome and teaming benefits.

