



Consultants



Testing watch outs with Agile Development





Testing watch outs with Agile Development

Agile development environments pose unique challenges for software testing, beyond the obvious issues like development speed, lack of documentation and planning horizon that does not extend past a couple of weeks. In our extensive experience supporting Agile teams with testing, we found that managers should anticipate and plan for a variety of challenges, including:

1. People are fallible

Compared to more structured Waterfall development, success with Agile is more dependent on the productivity, values and habits of the people participating. You need a very good and aligned team working at a fast pace in a collaborative fashion. This state, unfortunately, is never guaranteed particularly if the individuals are inexperienced, pursue different personal agendas or have poor interpersonal or communication skills. When this happens, the chances of testing (ultimately quality) problems occurring magnify.

Some fixes:

- Get the right people in the right roles
- Foster rich and timely communications plus a culture of continuous learning
- Make 'quality' a team goal, not just for individual testers

2. Assuming working code is a test pass

A large, working set of unit tests can lull the team into thinking the code quality is good. This would be a mistake. The tests could have worked as designed, but that does not presuppose the test design was not flawed in the first place. Flawed test designs can hide more fundamental quality issues and make future debugging more difficult and expensive.

Some fixes:

- Reconfirm the test design to ensure it is appropriate for the tasks
- Don't wait till the end to begin user acceptance testing
- Periodically run test reviews to ensure adequate test coverage and completeness



3. Code throughput complicates testing

One of Agile's great strengths – creating lots of code, fast – can quickly lead to major testing problems. In general, the more code that is delivered (with corresponding fixes made) the greater the code complexity and likelihood of important bugs slipping through. For example, a good set of happy-path tests in a single iteration could be a poor choice later on. We have also seen test scripts fossilize making them unusable for later iterations. Finally, program complexity increases the possibility of unintended consequences such as new bugs emerging or old bugs reappearing.

Some fixes:

- Minimize iterations by writing good code the first time
- Be realistic: with Agile's many benefits come some testing challenges
- When it comes to bugs, plan for the unexpected throughout the process

4. Testing requirements vs delivery goals

Agile best practices say that testing and development should be done in the same iteration in order to maximize quality and minimize rework. The reality of many projects is the opposite due to time, technical complexity and manpower constraints. Code is often delivered too late to the testers. All too often, the result is deferred or skipped testing, leading to decoupled team members and a breakdown in the Agile process.

Some fixes:

- Pair up testers and developers to ensure alignment and symbiotic workflows
- Prudently use automation and continuous integration to accelerate code development
- Bring in outsiders early so they are up to speed on the project from kick off

5. Forgetting Agile principles

We have found that in hybrid or even pure Agile environments team members will undertake testing, that while suitable for Waterfall methodologies, are clearly unsuitable for the project goals. For example, independent or solo testing activities (separated from development objectives) can slow down a collaborative team effort and provide narrow amounts of feedback. In other cases, diagnosing and recording every bug for a future fix may be a comprehensive approach to testing but not practical in an Agile environment. With Agile



projects it is more efficient to spend the extra time fixing the bug with the developer at the time than waiting till later.

Some fixes:

- Realign team around Agile goals and practices as well as business needs
- Pair developers and testers to accelerate the bug identification and resolution loop
- Be pragmatic around documentation creation and metric collection

Despite the challenges, Agile environments are still conducive to high quality testing. You just have to know how to do it. It is up to the testers, along with their developer counterparts and external partners, to ensure their practices and work styles are consistent with Agile methodologies.

All Rights Reserved ©.

For further information please contact:

Harry France
Sr. VP Sales
(519) 575-0314
hfrance@qaconsultants.ca