Agile Testing - Resource Guide

Purpose of This Document

This document is a guide to suggested resources and further reading on agile testing. We provide as many as possible of these via itslearning as PDF of papers and links to webpages. These resources complement the video lecture on 'lean testing'.

The Role of The Tester

One of the most influencial books on agile testing is Lisa Crispin and Janet Gregory's *Agile Testing - A Practical Guide for Testes and Agile Teams* [1]. Both authors have many years as agile testers, and the book is written from a practical perspective.

It is clear that testing is a key activity in agile development practices. What is less clear is the value that is added by having a testers — as well as programmers who test — as part of the agile team. Crispin and Gregory argue that there is benefit of having a testing role in the team that undertakes activities such as:

- working as part of the customer team to elicit requirements (in the form of examples and stories) and to understand the customer viewpoint;
- working as part of the development team to create functional tests from customer examples, to automate testing, to design for testability, and to build an infrastructure to support testing;
- fostering a "whole team" approach to quality — this contrasts with traditional development practices in which testers (or a QA team) have the responsibility for quality but little control over the development process and thus tend to act as 'gatekeepers'.

Both authors have active blogs:

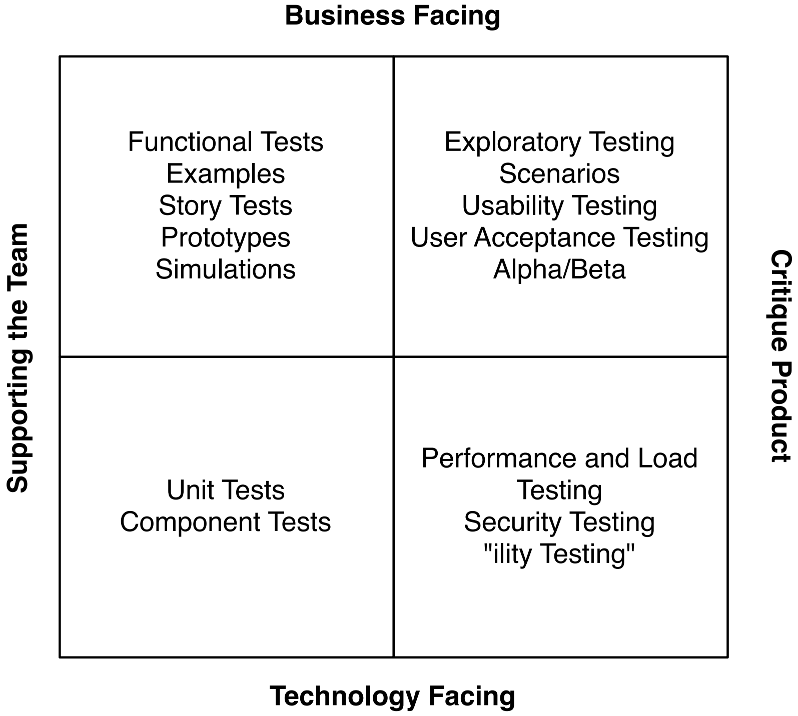
- http://lisacrispin.com
- http://janetgregory.ca

Suggested resources on the role of testers in agile teams, and transitioning from traditional to agile practices:

- Lisa Crispin's slidesets: "Are Agile Testers Different?" [2], "An Iteration in the Life of an Agile Tester" [3], and "Crossing the Chasm: Helping Testers Make an Agile Transition" [4]

- Lisa Crispin and Janet Gregory's article, "What's a Tester without a QA Team?", Agile Connection community, 2010 [5]

Agile Testing Quadrants



Business Facing

Supporting the Team

Critique Product

Functional Tests
Examples
Story Tests
Prototypes
Simulations

Exploratory Testing
Scenarios
Usability Testing
User Acceptance Testing
Alpha/Beta

Unit Tests
Component Tests

Performance and Load Testing
Security Testing
"ility Testing"

Technology Facing

Figure 1 — Agile Testing Quadrants (adapted from [1])

Crispin and Gregory’s book has popularised the agile testing quadrants that have proved a useful way to think about the types of testing that agile teams should be doing. The quadrants were first proposed by Brian Marick in a blog post [6], and extended by Crispin and Gregory.

Technology Facing & Supporting the Team (lower left quadrant)

This quadrant represents testing defined by the programmers that verifies the functionality of individual units or components in the system in order to ensure the internal code quality.

In many agile practices, this type of testing plays a central role through the technique of test-driven development (TDD). In TDD, the developer writes a test for a piece of functionality, and then develops the code to pass that test.

Business Facing & Supporting the Team (upper left quadrant)

This quadrant considers testing that — in contrast to unit/component testing — considers functionality at higher level that is meaningful to the customer, and with the purpose of demonstrating that the system has the functional behaviour of the system is consistent with the requirements.

The tests in this quadrant may be derived from the user stories and examples provided by the customer team. Crispin and Gregory avoid the use of “acceptance test” that is often used for these type of tests on the basis that tests in other quadrants may also be used for this purpose.

In addition, Crispin and Gregory consider prototypes etc. used as part of the dialogue with the customer to belong to this quadrant - even mock-ups of, for example, a GUI on paper.

Business Facing & Critique Product (upper right quadrant)

This quadrant encompasses manual testing that ensures that the system is really what the customer wants rather than simply meeting the team's understanding of what the customer requires. Crispin and Gregory argue that exploratory testing — unscripted and unconstrained testing guided by strategy and experience — is a core form of testing in this quadrant.

Technology Facing & Critique Product (lower right quadrant)

This quadrant considers types of tests related to non-functional aspects of the system such as security, performance, ability to handle load, scalability, reliability, safety, maintainability, interoperability etc.

Suggested resources related to agile testing quadrants:

- Lisa Crispin's slideset "Agile Test Planning with the Agile Testing Quadrants" [7]
- Gojko Adzic's blogpost "Let's break the Agile Testing Quadrants" [8] - argues that the horizontal separation between supporting the team and critiquing the product have become increasing blurred.
- Duncan Nisbet's blogpost "Dissecting The Testing Quadrants" [9] - explores the quadrants in more detail

Test Automation and Tools

The advantages of automating testing are particularly relevant in an agile context. For example, automated tests support techniques such as TDD, can provide timely and frequent feedback, and free up the programmer team to apply their knowledge elsewhere.

Crispin and Gregory argue that many of the tests that support the team — the unit/component tests (lower left quadrant) and functional ("acceptance") tests (upper left quadrant)— are amenable to automation. The automation of these tests will ideally lead to a continuous integration and build process. For unit/component tests, appropriate tools such as those in xUnit family. For functional tests tools needs to be able to describe the test in terms that the business understand as well as execute the test automatically; examples are Cucumber [10] and FitNesse [11].

Testing non-functional requirements — the types of testing in the lower left quadrant — will almost certainly require tools, for example to create a significant load on the system.

Suggested resources related to test automation and tools:

- Grigori Melnik and Frank Maurer's paper "Multiple Perspectives on Executable Acceptance Test-Driven Development" [12]
- Sean Stolberg's paper "Enabling Agile Testing Through Continuous Integration" [13]
- Eliane Collins and Vicente Ferreira de Lucena Jr.'s paper "Software Test Automation Practices in Agile Development Environment: An Industry Experience Report" [14]

- David Parson et al.'s paper “Influences on regression testing strategies in agile software development environments” [15]

References

[1] Lisa Crispin and Janet Gregory. Agile testing: A practical guide for testers and agile teams. Pearson Education, 2009.

[2] Lisa Crispin. Are Agile Testers Different? slideset from Agile Vancouver 2009. http://lisacrispin.com/downloads/AgileVancouverAgileTestersDifferent.pdf

[3] Lisa Crispin. An Iteration in the Life of an Agile Tester. slideset from Better Software 2008. http://lisacrispin.com/downloads/IterationInLife.pdf

[4] Lisa Crispin. Crossing the Chasm: Helping Testers Make an Agile Transition. slideset, 2008. http://lisacrispin.com/downloads/AgileVancouverCrossingChasmV2.pdf

[5] Lisa Crispin and Janet Gregory. What's a Tester without a QA team, Agile Connection community website, 2010, http://www.agileconnection.com/article/what's-tester-without-qa-team

[6] Brian Marick. My Agile testing project. http://www.exampler.com/old-blog/2003/08/21/#agile-testing-project-1

[7] Lisa Crispin. Agile Test Planning with the Agile Testing Quadrants, slideset from ADP Testing Workshop, 2009. http://lisacrispin.com/downloads/AdpTestPlanning.pdf

[8] Gojko Adzic. Let's break the Agile Testing Quadrants (blogpost). Oct 2013. http://gojko.net/2013/10/21/lets-break-the-agile-testing-quadrants/

[9] Duncan Nisbet. Dissecting The Testing Quadrants (blogpost). Feb 2014. http://www.duncannisbet.co.uk/dissecting-the-testing-quadrants

[10] Cucumber (website). http://cukes.info

[11] FitNesse (website). http://www.fitnesse.org

[12] Grigori Melnik, Frank Maurer. Multiple Perspectives on Executable Acceptance Test-Driven Development. Agile Processes in Software Engineering and Extreme Programming. Lecture Notes in Computer Science Volume 4536, 2007, pp 245-249

[13] Stolberg, Sean. Enabling agile testing through continuous integration. Agile Conference, 2009. AGILE'09. IEEE, 2009.

[14] Eliane Figueiredo Collins and V. F. de Lucena. Software test automation practices in agile development environment: An industry experience report. Automation of Software Test (AST), 2012 7th International Workshop on. IEEE, 2012.

[15] David Parsons, Teo Susnjak, and Manfred Lange. Influences on regression testing strategies in agile software development environments. Software Quality Journal: 1-23, 2014.