"Charting the Course ...

... to Your Success!"

Agile Testing and Software Quality

Course Summary

Description

As the various Agile methodologies have become common mainstream software development processes, software testers and QA staff are faced with the challenge of shifting from the traditional QA model to being an integrated part of an Agile team. While the standard testing tools, processes, techniques and methods are essentially unchanged, the are applied in often radically different ways as testers are integrated into Agile teams and take on roles and responsibilities that are new and outside the scope of the traditional tester role. Instead of evaluating products, testers find themselves writing tests to evaluate the development process and drive new sorts of processes like test driven development.

This course is designed to show software testers and QA analysts how to migrate their expertise and skills in testing and software quality into an Agile environment, partly by understanding the new roles and functions their work supports and also by learning to think about testing from an Agile perspective, applying Agile concepts and ideas to leverage their existing skills in an Agile environment.

The course begins with an introduction to Agile development emphasizing the Agile principles that underpin the different Agile approaches. These concepts are related to more traditional models of quality assurance and testing maturity such as Beizer's tester maturity model and the CMM test maturity models. This frames the role of the Agile tester in terms of more traditional testing excellence models actually being used to support the same Agile principles that are used by Agile developers. The modern concepts of software quality, good enough quality, risk mitigation, and related quality issues are also reframed in an Agile context without losing any of their effectiveness.

The standard Crispen and Gregory four quadrant model of Agile testing is presented showing how the standard testing methodologies map into the various quadrants, followed by detailed explorations of the test methods, automation tools and quality objectives in each of the quadrants. The practical issues of migrating from a standard test environment to an Agile environment are evaluated from both role and task perspectives.

One of the defining characteristics of Agile testing is the reliance on automation. The course looks at some of the various automation tools but also looks at the meta-testing problem of testing the automated testing tools.

The course concludes with a "A Day in the Life" which follows a typical tester through a simple Agile project to demonstrate the roles they play and the interactions they have with the other team members, as well as looking at the specific activities they engage in at each stage of the project from inception to final delivery.

Topics

- Understanding Agile Development
- Testing Maturity
- Agile Testing
- Software Quality
- Agile Testing Quadrants
- Q1: Technology Facing / Team Support
- Q2: Business Facing / Team Support
- Q3: Business Facing / Product Critique
- Q4: Technology Facing / Product Critique
- Moving to Agile Testing I
- Moving to Agile Testing II
- Test Automation
- Agile Testing Throughout the Project

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically

Agile Testing and Software Quality

Course Summary (cont'd)

Audience

The course is intended for software testers and test mangers or any other QA analysts who have an interest in Agile testing.

Prerequisites

The course assumes a working knowledge of testing concepts and techniques. Students who are unfamiliar with testing concepts and methods may have difficulty with some of the material.

Duration

Three days

Agile Testing and Software Quality

Course Outline

I. **Understanding Agile Development**
   A. Agile versus traditional development
   B. The Agile Manifesto and Agile principles
   C. Iterative development
   D. Different Agile approaches.
   E. Challenges for testing in an Agile environment

II. **Testing Maturity**
   A. Process maturity: the CMM model
   B. Beizer's levels of testing maturity
   C. Burnstein et al testing maturity levels
   D. The importance or testing maturity for Agile testing
   E. Testing throughout the application life cycle

III. **Agile Testing**
   A. What it means to do Agile testing
   B. Differences between Agile testing and traditional testing
   C. Working within an Agile team environment
   D. Testing as a continuous activity
   E. Testing as a driver for development
   F. Test Driven Development (TDD)
   G. Acceptance Test Driven Development (ATDD)

IV. **Software Quality**
   A. Good Enough Quality (GEQ) in Agile
   B. Risk, quality goals and management
   C. Planning testing to meet quality goals
   D. Using testing to make development efficient and effective
   E. Developing an initial Agile test plan
   F. The Agile tester's mindset

V. **Agile Testing Quadrants**
   A. Axis: Supporting the team vs critiquing the product
   B. Axis: Business facing versus technology facing
   C. Mapping traditional testing into the quadrants
   D. Test automation
   E. Quadrant specific test objectives

VI. **Q1: Technology Facing / Team Support**
   A. How tests drive development
   B. Unit tests versus integration tests
   C. Tests as a basis for collaboration
   D. Integration of testers into an Agile team
   E. Testers as quality consultants
   F. Agile roles as opposed to formal roles
   G. Automation and tools
   H. Deep dive into quadrant I

VII. **Q2: Business Facing / Team Support**
   A. Mitigating risk
   B. Developing testable requirements
   C. Domain testing – creating precision and clarity
   D. Slicing and chunking – incremental testing
   E. Developing exit/pass/fail criteria
   F. Eliciting examples and requirements
   G. Checklists, spreadsheets, mindmaps, mockup, and other tools
   H. ATDD and example based testing
   I. Functional and story based testing
   J. Automation and tools
   K. Deep dive into quadrant II

VIII. **Q3: Business Facing / Product Critique**
   A. Usability testing
   B. Exploratory testing
   C. User acceptance testing
   D. Alpha and Beta testing
   E. Exploratory testing automation
   F. Deep dive into quadrant III

IX. **Q4: Technology Facing / Product Critique**
   A. "ility" testing: reliability, maintainability.. etc
   B. Nonfunctional and performance testing
   C. Security testing
   D. Deep dive into quadrant IV

Agile Testing and Software Quality

Course Outline (cont'd)

X. Moving to Agile Testing I

A. Moving from separate QA teams to being embedded in the development team
B. Becoming process oriented instead of product oriented
C. Dealing with cultural and management issues
D. Defining the role and contributions an Agile tester makes

XI. Moving to Agile Testing II

A. Rethinking processes and "just enough" testing
B. Defect tracking and continuously updated quality metrics
C. Setting testing metrics for tester performance
D. Automation for defect tracking and testing documentation
E. Agile test plans, traceablility and test strategies
F. Consider testing everything
G. Test audits, reports and evaluation Agile style

XII. Test Automation

A. What can be automated in each quadrant
B. What should never be automated
C. Difficult to automate types of tests
D. Automation blind spots, things that are missed
E. Metatesting: testing the test automation tools

XIII. Agile Testing Throughout the Project

A. This module walks through a typical Agile project from beginning to end and examines the activities, roles and tools an Agile tester would use and how the tester would adapt the testing to evolving and changing conditions and circumstances.