The Agile Tester - Introduction

Agile Testing with
Acceptance Test Driven Development
and Behavior Driven Design

Two Day Course Overview

About the instructor.

Tim Walker is a career software engineer, educator and team coach applying "outside-in" behavior and test driven approaches. His goal is to assist agile teams achieve excellent results by "baking quality in", while constantly improving their the agile and lean development processes and results. Bringing over 30 years of iterative software engineering experience using Object Oriented Analysis and Design as well as Domain Driven Design to help teams build a "ubiquitous language" in to their test first disciplines. Developed and delivered "Executable Requirements with FitNesse" as a staff member of the Agile University and as an education director with an international consulting team. Developed, delivered and coached Agile Mastery, Agile Estimation and Planning and much more. Developed the "Scrum Accelerator" and delivered it to Agile 2008 in Toronto receiving several "best in show" evaluations. Have taught test first processes to many fortune 100 and 500 companies. Recently applying behavior driven and test driven development on multiple large scale utility projects managing “Smart Grid” devices and most recently on a mission critical government accounting system where we worked with the Relish authors to publish executable specifications as the source of truth for the API's and complex business rules.

Specialties: Lean and agile software testing, iterative processes applied at the enterprise and mission critical levels, Executable Requirements, Physical Device Integrations, Testing in the Four Quadrants, Ruby on Rails, Cucumber, BDD, TDD, OOAD, Scrum, Kanban, Java, C and Assembler.

Tim is a certified Scaled Agile (SAFe) Program Consultant (SPC).

The Agile Tester, LLC
Tim Walker
2537 Sweetwater Circle
Lafayette, Colorado 80026
(720) 323-4652
walketim@gmail.com
Twitter: @theagiletester, @walketim

Day 1 Morning

Introduction

An introduction and easy discussion about what and how will follow.

Introduction Lab Prep

We create our lab teams consisting of the business owner/SME, developer/tester, creating a name and a mock business and domain name.

Module 1 Introduction to Agile Testing

While agile and lean development practices have been evolving and becoming common place we find that we can improve our agility significantly by adopting agile testing approaches.

Module 2 Technical Debt in Agile

A lot of teams struggle to sustain agile and iterative processes in practice. Automation lags, quality is low and change is difficult. We explore why that happens and what the true cost is.

Module 3 Specification" or "Test"?

Teams to invest in the creation of requirements and test cases and then have issues with change management and maintenance. We explore the power of executable specifications.

Module 4 ATDD, BDD and TDD Explained

The class learns about Acceptance Test Driven Development, Behavior Driven Development and Test Driven Design technical practices and learns how and why these practices support agility.

Module 5 BDD in the Agile Context

Despite incorporation of scrum and other agile approaches teams still take a "v-model" approach to development. We look at what happens when we write executable specifications.

Module 6 New Role for QA

Without changing our approaches, traditional regression testing is just too slow to support rate of change required. Whole team approaches to quality and powerful exploratory testing emerge.

Lunch

The Agile Tester - Introduction

Day 1 Afternoon

Module 11 Introduction to Domain Driven Design

Domain Driven Design works with the principles established by Eric Evans to manage complexity in software development. We will connect the dots with our executable specifications.

Lab 11 Introduction to Domain Driven Design Lab

We will apply domain modeling to our business context and discover the ubiquitous language for our domain.

Module 7 Specification by Example

We will cover ambiguity in requirements and how to resolve it while providing a deeper shared understanding of them by expression with concrete examples.

Lab 7 Specification by Example Lab

We will explore some ambiguous requirements and work together in teams to improve them by incorporating examples.

Module 8 Mastering Cucumber

We will learn how to express requirements in the “Gherkin” language and how Cucumber is used to automate them.

Demo Cucumber Demo

The instructor will do a live demonstration using Cucumber to write and execute some simple behavior.

Lab Mastering Cucumber: Writing Cucumber Lab

The class will get hands on experience writing Gherkin to flesh out the acceptance of some given user story contexts. The teams will review their solutions with the class.

Adjourn Thank you!

Day 2 - Morning

Review **Review what we've learned**

Module 9 **Cucumber Best Practices**

The class will learn about a dozen common “smells” of poorly crafted Cucumber as well as the challenges they create. Best practice guidelines are offered.

Lab 3 **Rate My Cucumber Lab**

The class will take several real world cucumber examples and “sniff” them for the common smells and make recommendations for improvements.

Module 10 **Continuous Integration**

One of the reasons test first development practices increase agility is because we get feedback sooner. We will explore the importance of fast feedback and how we'll use continuous integration.

Module 12 **Publishing Tests: Single Source of Truth**

Our BDD practices create a unique and powerful source of truth of how our software works by publishing them as first-class documentation which supports searching, navigation that is always up to date.

Module 13 **Non-Functional Requirements and BDD**

BDD to used to support powerful system capability testing (performance, capacity, security etc.) of the software from the beginning of the software project ensuring constant releasability of the solution.

Lunch

Day 2 - Afternoon

Module 14 Agile Test Orchestration

It is critical to understand the agile testing quadrants and the testing pyramid to drive our focus in a strategic test plan and when to test through the user interface and where manual testing fits in.

Module 15 Evolutionary Database

In order for agile and iterative development to work we need a way to evolve our databases incrementally. Traditional mechanisms for managing databases are fragile and risky.

Lab Option The Cucumber Client Simulation

If the situation allows we will take some use cases from the customer domain and break into small teams to craft them with Cucumber's Gherkin. We will ask the team to think about the data required in setup and other tactical problems.

Lab Option The Cucumber Business Simulation

Otherwise we will take the business concept through very abbreviated 5 levels of agile planning. We'll take the top 5 stories discovered by the process and express the acceptance criteria for them in the first sprint applying what we've learned.

Adjourn Thank you.

LAB PREP

- The class will self organize in to teams of three.

- Write the words: QA, DEV, BIZ on pieces of paper, fold and place in a hat or similar.

- Each team member picks a paper and fills out a name placard, sticker or neck hanger with that role. This is the role they will play on the teams for the labs.

- Each team decides on a name and industry of a mock company or they can choose to use their current company. This will be used to drive the exercises in the labs.

- See lab handout for more information.

Agile Testing with
Acceptance Test Driven Development
and Behavior Driven Design

For any questions or to receive more information please contact:
tim.walker@theagiletester.com or twitter: @theagiletester