www.iconagility.com
(636) 530-7776

Agile Testing with Acceptance Test Driven Design (ATDD) and Behavior Driven Development (BDD)

In this 2-day course you will learn how to build quality into your product, while controlling scope and avoiding duplicated efforts. We will learn how to apply a 'whole-team' approach to quality and how to orchestrate feedback from your tests in order to be extremely effective. We will learn why the traditional approaches to test automation don't yield the returns we require and why these approaches do not increase quality despite, being expensive and costly to maintain.

This course is implemented as an interactive workshop aiming for about 50:50 lecture to lab ratio. Students will have fun, be energized and ready to apply this skill upon completion of this extremely engaging workshop.

We want product owners or non-technical business stakeholders, developers and tasters, as well as anyone interested in learning the craft of Behavior Driven Development (BDD) to attend.

Why?

The pattern repeats itself every day. Many teams implement Scrum or other iterative practices in their quest to be Agile. Initially it appears to work great, as the team can just go fast. But, just as quickly as they got started, they discover that they can no longer go fast—that the architecture has devolved, the code is a mess and the team starts to discuss "technical debt" as the reason they are no longer as Agile as they could be.
The reasons for this are many, but if we apply the same practices we've always used, we'll find ourselves in trouble sooner, rather than later.

And yet we know that some teams are developing complex software and deploying quality releases. In some cases, many times a day. So, what is it that these teams know and other teams wish they knew?

Objectives:

- Learn how agile practices create technical debt
- Understand how to "build quality in"
- Establish a whole-team approach to quality
- Learn how to control scope through a shared understanding of it
- Understand how to create an effective strategy for quality in our agile and lean practices

Audience:

All members of the cross-functional agile and lean development team. Especially suitable for product owners, developers and testers. Excellent way for engineering and project managers to understand quality in an iterative development model.

Pre-requisites:

Some experience with Agile and other iterative development models.

Duration:

2 days

Outline:

1. **Introduction to Agile Testing**
   - What is Agile Testing
   - Why we need approaches that support our agile practices
   - What we need to do to sustain agility and respond to change

2. **Technical Debt and Agile**
   - What is Technical Debt?
   - How do modern practices create Technical Debt?
   - How does Technical Debt affect our Agility
   - What can we do to eliminate Technical Debt?

3. **"Specification" or "Test"**
   - What is the difference between a requirement and a test?
   - How can we achieve a shared understanding?
   - How will we automate?
   - How we can remove ambiguity and create testable precision in our Agile models.
   - How a powerful Single-Source-Of-Truth emerges when we use tests as specifications
   - How will we implement this in our iterative model?

4. **ATDD, BDD and TDD Explained**
   - What are good working definitions for ATDD, BDD and TDD
   - How do we use them in our Agile processes?
   - How we create different kinds of automation for different "feedback requirement" needs
   - What is the role of the "tester" on a cross-functional agile team in the context of these practices?

5. **Behavior Driven Development in Agile**
   - What ate the most common pitfalls of an agile development process?
   - How can Behavior Driven Development help our agile processes?
   - How do we get started with BDD?
   - What are the artifacts of BDD?
   - How does BDD reduce Technical Debt?

6. **New Role for QA**
   - Why Testers should be first-class citizens of the cross-functional agile team.
   - How we can evolve our skills to support modern development practices.
   - A change model from traditional QA.

7. **Introduction to Domain Driven Design**
   - How can we use the Domain Model to help us in our Agile practices?
   - Why keeping the Domain Model at the core of the process helps us make good decisions.
   - How an understanding of the Domain Model and the "ubiquitous language" which evolves during BDD is powerful.
   - How creating context helps us communicate our Domain Model simply and at the right level for understanding.

8. **Specification by Example**
   - How does the use of concrete examples remove ambiguity and create testable, shared understanding?
   - How does Specification by Example help reduce waste and allow us to sustain agility?
   - How do we know when we're done?
   - How can we have confidence that the behavior that was developed in iteration 1 is still working in iteration 7?
   - How to conduct a "Specification Workshop"

9. **Mastering Cucumber and Cucumber Demo**
   - How do BDD Tools work?
   - How do I express specifications in Gherkin (the "language" of Cucumber
   - What is the syntax of Gherkin and how do I use it?
   - How do I use Cucumber to drive my automation and create knowledge about what is being checked?
   - How do we get started with Cucumber?

10. **Cucumber Best Practices and "Rate My Cucumber" Experience**
    - This is what we can we do to write good specifications and why this is so important.
    - How to communicate effectively with BDD.
    - Using our examples to communicate with other teams.
    - Why publishing or Gherkins is so Important.

11. **Continuous Integration**
    - Why it is critical to get fast-feedback from our development activities.
    - What is Continuous Integration (CI)?
    - How to get good reports from your CI that can be used to communicate progress?

12. **Publishing Tests: Single Source of Truth**
    - How do we engage the business by publishing our "cukes"?
    - How do we use our "Single-Source of Truth" to communicate with others and maintain and build a shared understanding?

13. **Agile Test Orchestration**
    - How do we orchestrate our test automation?
    - What frequency of feedback is appropriate for what kind of automation?

14. **The Cucumber Business Solution**
    - We will build, throughout the course and workshops that culminates in the simulation of "Specification By Example"

15. **Client Specific Simulation (Optional)**
    - In this simulation we will explore the client solution in groups using set-based approaches to create a shared understanding.