

Mastering Test First Development in Java with Cucumber and Friends

Course Summary

Description

This one-day class will get developers and testers "up and running" with "outside-in" practices using Cucumber and Junit in Java. This workshop is complementary to the two day "Agile Testing with Acceptance Test Driven Development and Behavior Driven Design".

This is a hands-on class working with requirements and driving code design making the connection between the natural language requirements and the system under test. This knowledge will allow teams to improve their agility by "baking quality in" to their projects as they develop it in our SAFe/Scrum/Kanban agile process methodologies.

Students will learn essential skills through hands-on experiences from installing, configuring, running and developing software with these approaches. The methods learned will be suitable in any development environment or technology stack, though Java will be used in this workshop. We will discuss integration with the popular IDEs and will support experimentation in the environments the students require in their day-to-day activities.

Objectives

By the end of this course, students will be able to:

- Develop software "outside in" using test-first approaches
- Create the unit, integration and functional automation driven from executable specifications BDD/ATDD through the TDD process
- Allow the team to build just the right thing while ensuring quality for continuous delivery and DEVOPs practices
- Create the right focus on a balance of test automation
- Increase the quality of the application design
- Find a testable solution which emerges organically

Topics

- Installing the tools
- Running from command line
- "Hello (Behavior Driven Design) World"
- File Structure
- Regular Expressions
- Passing Values
- Dealing with Tables
- Transforms
- Tags
- Maven and Gradle
- IDEs
- TDD and "Outside-In" development

Audience

This course is ideal for software developers or testers in any discipline.

Prerequisites

It is highly desirable that the student has attended the companion two-day course but this is not a strict requirement if students are comfortable with and have a basic understanding of Java development in an IDE as well as the basic command line. Concepts or experience regarding Behavior Driven Design, Cucumber and Test-First practices are helpful. Basic regular expression experience will be helpful.

Duration

One day

Mastering Test First Development in Java with Cucumber and Friends

Course Outline

I. Installing the tools

As a class, we will install the tooling to develop applications using the "outside-in" approach. Students will be required to bring their laptop and a workshop spirit to work with our trainers to get an environment up and running.

II. Running from command line

As a "building block for understanding" we will develop software running the tools from the command line. This will give the students a rich experience and knowledge suitable for transference to the needs of their technology stack.

III. "Hello (Behavior Driven Design) World"

Once we have a running environment we will use it to create executable specifications in "Gherkin" which can be run by Cucumber and see how we use "glue-code" to integrate natural language (in the language of the business) with the SUT (System Under Test).

IV. File Structure

Students will learn a basic folder structure required to support "tests as specification" right alongside the code it validates. This approach will serve us well regardless of whether we're using Maven, Gradle or something else.

V. Regular Expressions

We will learn the power of regular expressions to help us keep our specifications rich and expressive. We will learn how to handle similar "Step Variations", "Optional Values", "Pluralization", and more.

VI. Passing Values

We will learn how to pass data from our "executable specifications" in to our tests which enables a great deal of power and reuse of specifications.

VII. Dealing with Tables

We will learn how the "tables" we pass to Cucumber can take virtually any format (with/without column headers, single row tables, etc.) as a clean and powerful way of passing complex objects to our executable specifications.

VIII. Transforms

We will learn how we can "transform" data in to an object to be used in our tests. For example, when passing currency, for example, the System Under Test could receive a "Money" object and how this transformation can happen simply and with no additional work on the part of the developer or tester.

IX. Tags

Students will learn how the use of "tags" create a powerful "virtual hierarchy" of specification that allows a great deal of control over how and when these tests are executed as part of a deployment pipeline.

X. Maven and Gradle

We will move our ability to develop from the command line in to one of these popular build orchestration tools.

XI. IDEs

Students will understand how to run these behavior tests from within the IDE and how plugins assist us with code completion, automatic templating, selection of a library of reusable test steps and more.

XII. TDD and "Outside-In" development

Students will experience the full-cycle of "Outside-In" development as a comfortable and rewarding way to develop software in a way that they'll feel uncomfortable developing software any other way.