Test Driven Development in Java

Duration:	3 days
Type:	intermediate

Description

This course introduces Java developers to the tools and techniques involved in Test Driven Development. The course is workshop based, with delegates spending the majority of their time applying the techniques they have learnt to sample applications.

Prerequisites

Delegates should be experienced and confident Java developers.

List of Modules

The Business Case for Unit Testing

- Classifying the different types of testing
- What role do developers play in testing?
- Problems with ad-hoc unit tests
- Advantages of systematic unit testing

Key Activities in Unit Testing

- Defining your intent through tests
- Writing just enough code to pass
- Adding tests and refining the code
- Testing up to the point of boredom
- Triangulating on hard problems
- Refactoring the code and tests
- Building a suite of test cases

The Role of TDD in Modern Software Design

- Is there such a thing as Test Driven Design?
- The role of TDD in traditional processes
- The role of TDD in agile processes
- Situations where TDD cannot succeed

Writing Basic Unit Tests with JUnit 4

- Declaring test classes and methods
- Setting up and destroying resources
- Testing error handling behavior
- Running tests from Ant scripts

Advanced Unit Testing with TestNG

- Re-running failed tests only
- Creating instances via factories
- Injecting data into test methods
- Adding timeout values to tests
- Running test methods concurrently
- Establishing dependencies between tests
- Organizing test methods into groups

Unit Testing and Mock Objects

- Real world classes have dependencies
- Encapsulated dependencies prevent testing
- Using dependency injection to enable testing
- Distinguishing the different types of Test Double
- Understanding Stubs, Spys, Mocks and Fakes

Creating Mocks with JMock and EasyMock

- Comparing the design of JMock and EasyMock
- Adding expectations to mock objects with JMock
- Pre-programming mock objects with EasyMock
- Testing sequences of method invocations

Unit Testing and Architecture

- Separating different layers of the architecture
- Ensuring that components can be isolated
- Mocking out remote components and services
- Strategies for adding tests to legacy code
- Situations where unit testing is unsuitable

Unit Testing and the Spring Framework

- Integrating unit testing into Spring based projects
- Using the Spring framework for functional testing

Unit Testing and Databases

- Abstracting the database using Hibernate and/or the JPA
- Writing unit tests to verify your data access layer
- How in-memory Java databases can speed up testing
- Using DBUnit to simplify database-driven tests

Testing the User Interface

- Unit testing Swing GUI’s with FEST
- Java based tools for testing Web Applications:
- Low level web testing with HttpUnit
- High level testing with HtmlUnit and JWebUnit
- Browser based testing with WatiJ and Selenium