# Agile Development in Java

## Course Summary

### Description

Agile principles, practices and processes offer a path to sustainable development for individuals, teams and organizations. For many developers who want to focus on their craft, however, it is sometimes difficult to get a view of Agile development that is not either focused on a project management perspective or just on the practice of Test-Driven Development (TDD).

For the Java developer, an overview of the larger Agile process landscape needs to be complemented with the practical side of software craftsmanship. This ranges from understanding how Scrum can be fine-tuned with Lean thinking to exploring Extreme Programming practices, such as TDD and pairing.

The Agile Development in Java course is aimed at Java developers who want to learn what Agile means for them. It introduces a number of common agile techniques and puts these into practice in labs and exercises in pairs and groups, before applying these over a series of mini-iterations. The workshop balances taught material with practice, introducing requirement techniques, lightweight modelling techniques, tracking and estimating approaches, design principles, testing practices and refactorings.

### Objectives

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

- Describe representative agile development processes and common practices
- Slice up requirements in terms of goals and estimate and plan against them
- Understand design thinking appropriate for responsive development
- Learn how to carry out TDD effectively
- Put concepts and techniques into practice during labs and in a small, iterative workshop

### Topics

- Agile Development
- Common Agile Approaches
- Software Craftsmanship
- Test-Driven Development
- Design Practice
- Goal-Structured Requirements

### Audience

The course is suitable for Java developers who wish to learn more about the practical side of Agile development, particularly TDD and incremental development.

### Prerequisites

There are no prerequisites for this course.

### Duration

Three days

# Agile Development in Java

## Course Outline

**I. Agile Development**
- A. Software development and change
- B. Agile values and principles
- C. Iterative and incremental development
- D. Visualization of progress
- E. Kicking off and closing out iterations
- F. The role of testing
- G. Modelling in an Agile context
- H. Plan
- I. Do
- J. Study
- K. Act

**II. Common Agile Approaches**
- A. Extreme Programming
- B. XP1 and XP2
- C. Scrum
- D. Scrum roles, events and artefacts
- E. The Nokia test
- F. Lean Software Development
- G. Lean principles
- H. Kanban for software
- I. Limiting work in progress (WIP)

**III. Software Craftsmanship**
- A. Code quality and development skills
- B. Elements of well-crafted code
- C. Coding guidelines benefits and pitfalls
- D. Code sufficiency versus overdesign
- E. Technical debt and code smells
- F. Refactoring
- G. Programmer testing

**IV. Test-Driven Development**
- A. Good Unit Tests (GUTs)
- B. Plain Ol' Unit Testing (POUT)
- C. Defect-Driven Testing (DDT)
- D. Test-Driven Development (TDD)
- E. Key TDD practices and the test-first cycle
- F. Behavioral testing based on propositions
- G. Negative test cases
- H. Overview of JUnit

**V. Design Practice**
- A. Agile architecture and responsive design
- B. Pattern thinking
- C. Class hierarchy design
- D. Acyclic dependencies
- E. Interface decoupling
- F. Transitive and external dependencies
- G. Test doubles
- H. Components with single responsibilities

**VI. Goal-Structured Requirements**
- A. Using use cases, scenarios and user stories
- B. Incremental development
- C. Lightweight use cases
- D. User story styles and guidelines
- E. Prioritization in terms of value and risk
- F. Scenario and task estimation
- G. Tracking