



☎ +44 7989 401397

✉ info@olsensoft.com

## Test-Driven Development using Cucumber (3 days)

### Course overview

Cucumber is the leading framework for TDD and BDD in Java, and it ties in well with a host of application types and Java technology APIs. You'll learn all about these integrations during the course.

### What you'll learn

- TDD and BDD essentials
- Achieving BDD using Cucumber
- Understanding Cucumber API and features
- Using Cucumber with databases, DI containers, and Web/REST applications
- Testing legacy code

### Prerequisites

- Familiarity coding and/or testing Java applications

### Course details

- **Introduction to TDD/BDD and Cucumber:** Getting started with behaviour-driven development; BDD vs TDD; Overview of Cucumber; How Cucumber works; Tooling up
- **Getting Started with Cucumber:** Creating features; Creating step definitions; Implementing step definitions; Testing our classes; Making assertions; Making the tests work
- **Getting Started with Gherkin:** Overview of Gherkin; Features; Scenarios; Comments; Spoken languages
- **Step Definitions External Perspective:** Steps and step definitions; Capturing arguments; Multiple captures; Achieving more flexibility; Returning results
- **Cucumber Best Practices:** Issues; Collaboration; Writing maintainable tests; Guidelines for preventing defects
- **Step Definitions Internal Perspective:** Domain model; Transforms; Custom helper methods; Dependency injection
- **Enterprise Development:** Asynchronous architecture; Message queues; Asynchronous components
- **Databases:** Interactive database development; Refactoring to use a database; Writing and writing to the database; Cleaning the database;
- **Dependency Injection:** Overview of DI; DI in Cucumber; PicoContainer; Guice; Weld CDI

- [Web Applications](#): Overview of Selenium WebDriver; Interacting with the DOM; Handling failures; Ajax
- [REST Web Services](#): Overview of REST; REST and Cucumber; Guidelines and best practices
- [Testing Legacy Code](#): The challenge of legacy code; Refactoring techniques; Mocking techniques; Additional useful guidelines and techniques