Test-Driven Development

🕘 Duration: 2 Days
🈂 Available Languages: English German

Audience

Software Craftsmen: Software Developers, Architects, Scrum Masters.

Precondition

Good knowledge of the programming language used.

Goals

Learn how to develop software using Test-Driven Development.

Contents

- Software Architecture Fundamentals for TDD
  - The two values of software
  - The ATP-Trinity of project code
  - The Importance Priorities: Automation > Test > Production
  - The four major design smells
  - Cohesion and Coupling
  - What is "testable" and how is it related to maintainability / Clean Code?
- Unit Testing fundamentals
  - The job of a test framework
  - How xUnit frameworks typically work
  - JUnit 3, 4, 5; TestNG
  - The Single-Assert Rule
  - The Test-Pyramid
- TDD fundamentals
  - The Three Laws of Test-Driven Development
  - The Red-Green-Refactor Cycle
  - The FIRST principles
  - How to Start
- Test Doubles (Stubbing and Mocking)
  - The Ontology of Test Doubles
  - The Two Schools of TDD: Stateism vs Mockism
- BDD – Behavior Driven Development
  - Unit Testing vs Acceptance Testing
  - Given-When-Then vs 4 A's
- TPP – Transformation Priority Premise
  - Transformation vs Refactoring
  - The Sequence for Tests
- Outlook

- Working Effectively with Legacy Code
- How to migrate Test-Last to Test-First
- Characterization Testing
- TDD and the SOLID principles
- TDD and Agile Development (Scrum, XP, Kanban, Lean)
- TDD and Software Craftsmanship
- TDD and Pair Programming - Ping Pong
- TDD and Continuous Integration / Continuous Delivery / DevOps

The course uses JUnit 5.4.2. Differences between JUnit 5, 4, and 3, and TestNG are covered in detail.

Booking

Contact Shweta Sadawarte, +91-99-22925990, training@nelkinda.com