

Clean Code, TDD, BDD Crash Course (C++)

 Duration: 2 Days

 Available Languages: English German

Audience

Product Owners, Business Analysts, Requirements Engineers and Software Developers.

Goals

Learn how to improve your Agile Process with Behavior-Driven Development.

Contents

- Clean Code Introduction
 - Clean Code and Scrum/XP
 - The Two Values of Software
 - The 4 Design Smells
- Clean Code Basics
 - Form and Names
 - Functions and Function Structure
 - Programming Paradigms Revisited
- Clean Architecture: Principles of Object-Oriented Class Design
 - Coupling and Cohesion
 - SOLID Principles
 - Package Principles
- Test Basics
 - Testing and Scrum/XP
 - Test Pyramid
 - Test Automation, Testing, and CI/CD
- Test-Driven Design
 - Anatomy of xUnit Frameworks; AceUnit
 - Definition of Testable
 - The Three Laws of TDD
 - The Red-Green-Refactor Cycle
 - Principles of TDD
- Advanced TDD
 - The Transformation Priority Premise, Test Sequence
 - Test-Doubles, Mocking with GMock
 - Stateism vs Mockism
 - How TDD drives an emergent design
- Behavior-Driven Development
 - The Gherkin Language, Feature Files
 - Integrating and Mapping BDD in Scrum/XP - Acceptance Test-Driven Development

- Cucumber for C++
 - Installing Cucumber C++
 - Setting Up Cucumber C++ with an IDE
 - Running Feature Files
 - Implementing Step Definitions
 - Assertions with Google Test

In open classroom training, we use a specific set of tools listed below. In brackets are alternatives to which we can tailor the training in case of in-house training.

Language

C++ (C#, Go, Java, JavaScript, Scala)

IDE

CLion (Atom, Eclipse, Goland, IntelliJ Idea, NetBeans, Visual Studio Code)

Unit Test Framework

AceUnit (Google Test, TestNG, JUnit 3, JUnit 4, JUnit 5, ScalaTest, NUnit, Visual Studio Testing Framework, Mocha)

BDD Framework

Cucumber (Godog, Fit, FitNesse, JBehave, JDave, Concordion, SpecFlow)

Booking

Contact Siddhesh Nikude, +91-95-52572354, training@nelkinda.com