NELKINDA SOFTWARE CRAFT

TRAINING

Agile Architecture and Design

🕘 Duration: 5 Days
Available Languages: English

Audience

Software Developers, especially principals, lead developers and architects; Software Craftsmen.

Precondition

Experience in software development; good knowledge of a programming language.

Goals

Learn the latest technologies in Agile architecture and design for both, Embedded Software Development and Full-Stack Development.

Contents

- Process Fundamentals
  - # The Spiral Model, iterative and incremental
  - # Overview of Scrum and XP
  - # Conway's Law - why Process and Architecture go hand-in-hand
  - # Continuous Integration and Continuous Delivery - Walking Skeleton and End-to-End
  - # The Design Smells: Rigidity, Fragility, Immobility and Viscosity
  - # The Two Values of Software
- Fundamentals of Object-Oriented and Object-Functional Programming
  - # Structured Programming
  - # Object-Oriented Programming
  - # Functional Programming
  - # Classes, Objects, Structures, Data Structures
  - # Functions / Procedures / Methods / Operations / Messages
  - # Visibility / Accessibility
  - # Encapsulation, Inner Classes, Inheritance
  - # Polymorphism, Higher-Order Functions, Closures, Lambda Expressions
- Agile Requirements Engineering
  - # User / Actor / Persona
  - # User Story, Use Case, UML Use Case Diagram
  - # Beyond User Stories: Epics, Goals, Features, Tasks
  - # UML Class Diagram
  - # Analysis vs Design
  - # The Gherkin Language and BDD
- Agile Testing

  - # Why Testing and Architecture cannot be separated
  - # TDD: The Three Laws of Test-Driven Development, Benefits, Red-Green-Refactor Cycle
  - # BDD
  - # Behavior and State
  - # Mocking
- Agile Architecture
  - # Coupling and Cohesion
  - # Domain-Driven Design
  - # The SOLID and Package Principles
  - # UML: Class Diagram, Package Diagram, Composite Structure Diagram, Component Diagram, Deployment Diagram, Profile Diagram, Use Case Diagram, Activity Diagram, State Machine Diagram, Sequence Diagram, Communication Diagram, Timing Diagram, Interaction Overview Diagram
- Patterns
  - # Design vs High-Level Design
  - # The Gang of Four Patterns, i.e. Template Method, Singleton, Abstract Factory
  - # So-called "Architecture Patterns", i.e. Micro Kernel, Multi-Tier, Micro Services, Protocol Stack

Examples are in UML 2.5, C11, C++11 / C++14, Java 8 and other languages. The C/C++ compiler used is GCC 5.2.

Booking

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