Software Craftsmanship

Working software is the primary measure of progress. And at the heart of any working software are the lines of code and its structure. A software craftsman takes responsibility and pride in the detailed works describing those lines and the beauty in its simple structure.

Course Overview

This 1-day experiential learning course will prepare you to become a Software Craftsman. It provides concrete experiments on Agile engineering practices with Craftsmanship mindset that are essential to write a clean, well-crafted code. It also presents in detail how software developer can take the most out of your technical tools (technology framework and/or IDE) to increase your productivity and confidence when working within a large code base.

Course Objectives

The course is entirely hands-on coding practices with different set of consecutive exercises which participants will attempt to solve with collaborative guidance from trainer. After each exercise, a debrief will be conducted to reinforce the learning objective.

Each exercise, example is used to assist the participants to develop the knowledge, skills, capabilities and insights needed to make the shift from the traditional software development practices to Agile approach.

At the end this course, trainees have a clear understanding about:

- Writing clean code that is easy for everyone (including testers, business analyst, Product Owner) to understand
- Principles and strategies when dealing with legacy code
- How to write Unit Tests and how not to write Unit Tests
- Value and mindset of a Software Craftsman

Targeted Audience

This course is intended for software architects, senior software engineers, developers, team leads, scrum master, and anyone who is expected to be part of the development team using agile.

Prior Knowledge and pre-requisites

Participants should have a good knowledge of software development and project life cycle. Participants are also required to:

- Bring their laptop with an installed IDE.
- OOP framework up and ready to code.
- A unit test framework with mocking library.
- Version Control System like Git is not mandatory but desirable.
Suggested Stacks
Java: Junit, Ant, Mockito
.Net: NUnit, XUnit, Moq

Course Content

Craftsmanship Overview
- Manifesto of Software Craftsmanship
- Craftsmanship in an Agile world
- Extreme Programming practices with
  Craftsmanship altitudes

Clean and Well-Crafted Code
- S.O.L.I.D principles explained and applied
- Unit tests and mocking frameworks
- Test Driven Development (TDD)
- Behavior Driven Development (BDD)
- Identify code smell and refactor

Working with Legacy Code
- Principles, strategies and tips
- Do and don’t when dealing with legacy code

Architecture and Design
- Emergent Architect with Simple Design and System Metaphor
- Domain Driven Design (DDD) for more flexibility and understandability
- Achieve Collective Code Ownership through code collaboration

Related Courses
- Agile Essentials
- Scrum Master with PSM I Certification
- Scrum Product Owner with PSPO I Certification
- Kanban & Lean Foundations
- Leading SAFe® 4.0 with SA Certification
- SAFe® 4.0 Advanced Scrum Master with SASM Certification

Trainers
Training will be conducted by an experienced trainer and Agile coach from why innovation.

Course Information
Each session will be limited to a maximum of 12 participants.

Duration: 1 day
Location: Singapore, Hong Kong
Training Fee: SGD 800 | HKD 4700