+33 (0)1 48 10 09 46
formation@ambient-it.net
140 Avenue Jean Lolive 93500 Pantin - France

Updated 10/10/2024

Sign up

Software Craftsmanship & Good Development Practices training

3 days (21 hours)

Presentation

Software Craftsmanship is an approach focused on excellence and continuous improvement in software development. This training course will enable you to advocate code quality and technical mastery. It will encourage you to engage in a process of constant improvement, integrating practices and principles that guarantee the robustness of projects. During the course, you will explore the fundamental concepts of Software Craftsmanship, such as history, evolution and the detection of technical debt. You'll learn to master SOLID principles and apply practices such as Clean Code, YAGNI, DRY and KISS. The program will cover advanced concepts such as Behaviour Driven Development (BDD) and its applications in the project lifecycle. You'll learn how to use frameworks like Gherkin and Cucumber to improve communication between stakeholders. Practical exercises such as Code Katas will enable you to consolidate your TDD and BDD skills, reinforcing your theoretical understanding with practical applications. Thanks to this training, you will have acquired solid skills in software development and project management. You'll be able to assess and reduce technical debt, apply Software Craftsmanship principles and use methodologies such as TDD and BDD.

Objectives

- Understand the fundamental concepts of Software Craftsmanship
- Mastering the detection and management of technical debt
- Apply SOLID principles and Clean Code practices
- Integrate test-driven development (TDD) into projects
- Adopt a culture of continuous improvement and team mentoring

Target audience

- Developers
- Architects
- Project managers

- Testers

Prerequisites

- Basic programming skills
- Experience in software development
- Knowledge of agile concepts

Software Craftsmanship & Good Development Practices training program

Day 1: Introduction

Part 1: Fundamental concepts

- Introduction to Software Craftsmanship: history and developments
- How can a project fail?
- The birth of Craftsmansip software
- Technical debt
  - How do you detect technical debt?
  - Controlling technical debt
  - Refactoring

Part 2: Software Craftsmanship

- Project lifecycle: Waterfall mode vs. Craftsmanship mode
- The right SOLID principles with illustrative use cases
- Understand the key principles: Clean Code, YAGNI, DRY, KISS

Day 2: Advanced

Part 1: Test-Driven Development (TDD)

- The idea of Test Driven Development
- TDD as a workflow
- Limits of TDD

Part 2: TDD in practice

- Concept of the Kata Code
- Code kata craftmanship sessions on Java exercises

Day 3: Advanced practices

Part 1: Behaviour Driven development (BDD)

- What is BDD?
- BDD practice in the project lifecycle
- Gherkin and Cucumber frameworks

Part 2: BDD in practice

- Installation of dev environments and frameworks
- Advanced craftmanship practical kata code session
- BDD practical kata code session

Part 3: Continuous improvement culture and coaching

- Software Craftsmanship in a team context: mentoring and coaching
- Code reviews: practices, tools and methodologies
- Personal development plan: how to progress as a Software Craftsman
- Conclusion and closing: open discussion on practical application in a professional context real

Companies concerned

This course is aimed at both individuals and companies, large or small, wishing to train their teams in a new advanced computer technology, or to acquire specific business knowledge or modern methods.

Positioning on entry to training

Positioning at the start of training complies with Qualiopi quality criteria. As soon as registration is finalized, the learner receives a self-assessment questionnaire which enables us to assess his or her estimated level of proficiency in different types of technology, as well as his or her expectations and personal objectives for the training to come, within the limits imposed by the selected format. This questionnaire also enables us to anticipate any connection or security difficulties within the company (intra-company or virtual classroom) which could be problematic for the follow-up and smooth running of the training session.

Teaching methods

Practical course: 60% Practical, 40% Theory. Training material distributed in digital format to all participants.

Organization

The course alternates theoretical input from the trainer, supported by examples, with brainstorming sessions and group work.

Validation

At the end of the session, a multiple-choice questionnaire verifies the correct acquisition of skills.

Sanction

A certificate will be issued to each trainee who completes the course.