

Updated on 10/04/2024

Sign up

# Clean Code training: Produce quality code

3 days (21 hours)

## Presentation

**Clean code** that's easy to understand and modify, so you can develop your structure or application quickly. That's the promise of Clean Code.

It's not a set of rigid rules to be respected, but an association of principles that reinforce the quality of your developments. The aim of Clean Code is to enable any programmer to easily edit and understand the code of your projects.

The benefits of implementing Clean Code within your organization include reduced errors, improved deliverability and greater collaboration.

Our Clean Code training course will introduce you to its various principles, such as the SOLID framework, good **naming**, structure and formatting practices, Test Driven Development (TDD) and Test Driven Development (BDD). Finally, we'll introduce you to best practices in productivity and collaboration.

At the end of this training course, your employees will be able to understand and implement a Clean Code approach within your organization.

## Objectives

- Understanding Clean Code, its principles and benefits
- Structuring and formatting your programs
- Understanding BDD and its usefulness
- Understanding TDD and its benefits
- Good programming productivity and communication practices

# Target audience

- Web and application developers
- Architects
- Director

# Prerequisites

Programming experience.

# Clean Code training program

## Introduction to Clean Code

- What is clean code?
- The good coder and the bad coder
- Why use Clean Code? Measure the cost of faulty code
- Key points of the Clean Code

## The SOLID principles

- History
- Liskov's substitution principle
- Dependency inversion
- Sole responsibility
- Interface separation
- The open-close principle
- Don't Repeat Yourself (DRY)

## Naming conventions

- Clear, simple names
- Avoiding ambiguity
- Names that can be searched for and pronounced
- Pitfalls to avoid
- Naming classes
- Naming methods

## Good code structure

- The importance of a good structure
- Data structure

- Object structure
- Vertical formatting
- Horizontal formatting

## The functions

- The structure
- Avoid side effects
- One level of abstraction per function
- Use the right arguments
- Use the right declarations
- Separate orders

## Classes and comments

- Organizing your classes
- The importance of commenting
- Good review vs. bad review
- Don't use comments to improve your code

## Test Driven Development (TDD)

- Why use TDD?
- Fundamental principles
- Clean testing
- The acronym FIRST
- Tools presentation
  - jUnit
  - Mochajs
  - Mockito
  - Jest
  - Pytest

## Behaviour Driven Development (BDD)

- BDD presentation
- Why automate your tests?
- BDD vs TDD
- Defining your project and prerequisites
- BDD tools
  - Behat
  - Zephys SScale
  - Easy B
  - JDave
  - JBehave

## Time management

- The benefits and risks of meetings
- When to meet and how?
- Keep your focus and avoid multitasking
- Time boxing
- Automate repetitive tasks
- Using IDEs

## Collaboration

- Using communication tools
- Assigning specific roles
- Good communication practices
- Sharing knowledge

## Companies concerned

This training 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.

