Course : Java, testing and quality

Practical course - 3d - 21h00 - Ref. JTS
Price : 1940 CHF E.T.

★★★★★   **5 / 5**

Software quality has become a key issue. You will use Open Source tools to industrialize your developments and guarantee this quality. You'll learn how to manage sources, documentation, anomalies and changes to an application, as well as all the associated tests.

◎ Teaching objectives

At the end of the training, the participant will be able to:

- ✓ Industrializing software development
- ✓ Manage development sources via a centralized or decentralized system
- ✓ Manage application bugs and functional upgrades
- ✓ Implement coding standards and software documentation
- ✓ Implement different types of tests (unit, functional, robustness and performance tests)
- ✓ Automate defined tasks

Intended audience
Java/Java EE developers, quality managers and project managers.

Prerequisites
Good knowledge of Java programming.

Course schedule

(1) General introduction

- What is software quality? Why do it? How to industrialize software development?

PARTICIPANTS
Java/Java EE developers, quality managers and project managers.

PREREQUISITES
Good knowledge of Java programming.

TRAINER QUALIFICATIONS
The experts leading the training are specialists in the covered subjects. They have been approved by our instructional teams for both their professional knowledge and their teaching ability, for each course they teach. They have at least five to ten years of experience in their field and hold (or have held) decision-making positions in companies.

ASSESSMENT TERMS
The trainer evaluates each participant's academic progress throughout the training using multiple choice, scenarios, hands-on work and more.
Participants also complete a placement test before and after the course to measure the skills they've developed.

2 Source management

- Definition, relevance to work organization, uniqueness of the development environment.

Hands-on work
Introduction to CVS and/or Subversion (SVN) open-source source management software.

3 Tool for managing bugs, anomalies and functional changes

- Functional description, interest for developers, users, managers.
- Definition and interest in sorting bugs, anomalies and functional evolutions.

Hands-on work
Presentation of Bugzilla and/or Mantis, free software for managing bugs, anomalies and functional evolutions.

4 Coding rules and documentation

- Interest (beyond clan oppositions). Beyond style: file organization rules...
- Interest in documentation, code documentation and other documentation (HOWTOs, etc.).

Hands-on work
Review of Sun's recommendations and coding rules. Presentation and configuration of CheckStyle. Code documentation exercises.

5 Tests

- Why test?
- Overview of the different types of testing: unit, functional, robustness and performance tests.
- Which tests to run and when?
- Usefulness of "Mock" and "Fake" objects during unit testing. Unit test coverage.

Hands-on work
Practice unit testing with JUnit (free software) and other tools. Introduction to functional testing with Mercury (commercial software) and HttpUnit (free software).

6 Delivery procedures

- Importance of delivery procedures. Classic and non-classic steps (manual and robustness tests).

7 Task automation

- Presentation of tasks to be automated.

Hands-on work
Presentation and introduction to Ant (free software) and/or Maven.

8 **General conclusion: quality measurements**

- Summary of quality measures: coding conventions, documentation, test coverage, procedure automation.
- Implementation of a quality dashboard.