Publication date : 03/01/2024

Course : Software test automation

Practical course - 4d - 28h00 - Ref. OTT
Price : 2850 CHF E.T.

4,1 / 5

Automating software development testing is a costly but profitable investment. Discover the organizational/financial aspects, best practices and tools for automating different types of testing: unit tests, integration tests, functional tests and performance tests.

Teaching objectives

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

- Use a test repository and set up links to related tools to automate a test campaign
- Perform a static analysis of code and test coverage and integrate it into a software production chain
- Use dummy object, stub or simulacrum techniques and set up a continuous integration server
- Implement functional test automation on fat clients and the Web
- Automate performance tests, including a test to determine the denial-of-service threshold

Intended audience

Developers, software quality and testing managers, test automation engineers, MOE project managers.

Prerequisites

Good knowledge of Java programming.

Practical details

Hands-on work

Theoretical sequences alternate with practical work.

Course schedule

PARTICIPANTS

Developers, software quality and testing managers, test automation engineers, MOE 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.

1 Introduction, reminders of the software testing process

- The role of testing in the development process.
- Tests: unit, functional, etc.
- The different test methods.
- Test process and test strategy.
- Tools and methods at different stages.

Hands-on work

Installation of a few tools.

2 Automated test management

- Managing requirements coverage through testing. Notion of coverage and granularity.
- Development process: organization of test suites and case creation.
- Should a test be automated? Criteria to take into account?
- Preparing for automation.
- Construction of the test population.
- Test development and verification (Review)
- Execution, recording of anomalies. Notion of incident report according to IEEE.
- Anomaly managers. Automate anomaly creation.
- Analyze test execution results. Test consolidation.

Hands-on work

Use of a test repository. Setting up bridges to related tools (TestLink/Squash/Quality Center or others...).

3 Unit test automation

- Organization and best practices for unit testing.
- Automation criteria.
- Unit testing: Tests Driven Development.
- Measuring code coverage: structural test coverage, instruction coverage and branches.
- Static code analysis: tool-based analysis of non-executing source code (coding rules): Checkstyle, Cobertura.
- Automation with a configuration file.
- Dynamic code analysis: coverage of instructions, branches, predicates, etc.
- Automation with a coverage analysis tool.
- Unit test organization, peer programming, peer testing.
- Using frameworks: test script management, test data management, results retrieval.

Hands-on work

Implementation of unit tests with Junit/TestNG. Integration into a software production chain (Ant/Maven...).

TEACHING AIDS AND TECHNICAL RESOURCES

• The main teaching aids and instructional methods used in the training are audiovisual aids, documentation and course material, hands-on application exercises and corrected exercises for practical training courses, case studies and coverage of real cases for training seminars.

• At the end of each course or seminar, ORSYS provides participants with a course evaluation questionnaire that is analysed by our instructional teams.

• A check-in sheet for each half-day of attendance is provided at the end of the training, along with a course completion certificate if the trainee attended the entire session.

TERMS AND DEADLINES

Registration must be completed 24 hours before the start of the training.

ACCESSIBILITY FOR PEOPLE WITH DISABILITIES

Do you need special accessibility accommodations? Contact Mrs. Fosse, Disability Manager, at psh-accueil@orsys.fr to review your request and its feasibility.

4 Automated integration testing

- Strategies specific to integration: big-bang, "as we go", incremental, etc.
- Bottom-up versus top-down integration. Mixed integration.
- Simulacrum objects: plugs to simulate called functions, mocking to replace an object. Frameworks.
- Continuous integration: configuration manager, build builder, management tools.
- Focus on a software configuration manager.
- Automatic fault reporting.
- Automatic, cyclical execution of software tests.
- Focus on a build manufacturer.
- Focus on a continuous integration server.

Hands-on work

Use of simulacrum objects. returnchariot Implementation of a continuous integration server (Jenkins, Hudson or Continuum).

5 Functional test automation

- Definition of functional testing, non-regression.
- The test simulates user actions based on user interfaces (GUIs).
- Findings on functional test automation.
- Test automation via the HMI, via programming interfaces (APIs).
- Tool chain, test robots, scripting (public APIs).
- Test obsolescence management.

Hands-on work

Automate functional tests on thick clients and the Web. Use of robots and a public software layer (API). Implementation of an automation tool (Katalon Studio with Selenium/HP QuickTest Pro).

6 System test automation

- Different types of system tests: performance, load, stress.
- Performance test findings.
- Performance tests, definition of indicators.
- Performance test scripts. Tool implementation and analysis.
- Load tests, stress tests: presentation of key indicators and tools.
- Manage test database. Context management.

Hands-on work

Set up a load test.

7 Summary

- Test automation: direct and indirect costs.
- Quantitative gains from automation: volume, modularity...
- Qualitative gains from automation: saved operating procedures, no errors, confidence...
- Measuring return on investment.

Dates and locations

REMOTE CLASS

2026 : 2 June, 15 Sep., 8 Dec.