Course : Automatic testing and continuous integration in C++

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

★★★½ 3,8 / 5

This course presents and implements an application testing process in a C++ environment. You'll see how each stage of the testing process can be automated. You'll learn how to check the effectiveness of your unit tests and write system tests. You will set up continuous integration.

Teaching objectives

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

- Set up the content and follow-up of the automation project
- Define test plan coverage and organize the test repository
- Implement specific tools to automate unit tests in C++
- Automate GUI, Web Service and scalability tests
- Mettre en place l'intégration continue et analyser le reporting des résultats

Intended audience
Software quality and test managers, developers, test automation engineers.

Prerequisites
Basic knowledge of C++ programming is required.

Practical details
Hands-on work
General tools: Testlink, Selenium, Mantis, Hudson or Jenkins, SVN, SOAP UI.

Teaching methods
C++-specific tools: Make, Gcov, Googletest (GTest) or CppUnit, AutoIT (windows), QTtest.

Course schedule

PARTICIPANTS
Software quality and test managers, developers, test automation engineers.

PREREQUISITES
Basic knowledge of C++ programming is required.

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 A reminder of the testing process

- Definition. Test levels: unit, integration, systems and acceptance, confirmation and non-regression.
- Different testing techniques. The testing process.
- Development and test cycles.
- Test strategy. The benefits of end-to-end strategies.
- Automation decision. What follow-up?

Storyboarding workshops
Definition of the tasks involved in successfully automating a C++ application.

2 Analysis and design

- Presentation of testing techniques.
- Requirements, test, test condition, test case, test procedure, prerequisites.
- What is a cover?
- Presentation of an open-source testlink tool.
- Best practices: coverage, valuation, test writing.

Hands-on work
Implementation of the test repository. Overview of open source and pay tools for test management, presentation of other tools (TFS).

3 Unit test automation

- Unit testing: what is a driver?
- Using plugs to automate tests.
- Coverage types: instruction, branch, condition, decision.

Hands-on work
Implementation of Google's unit testing framework in C++ (GTest) and a test plug. Code coverage verification with Gcov.

4 System test automation

- Organize test execution: test environments, anomaly manager.
- Automate system tests.
- API and WebService testing. GUI testing. Load testing.
- Best practices in automation.
- Introducing Selenium for the Web.
- Solutions for other types of HMI.

Hands-on work
Automated HMI/Web testing with AutoIT/Selenium/QTTest. Web Service test automation with SOAP UI. Implementation of load tests with SOAP UI.

5 Continuous integration

- Continuous integration: principles and benefits.
- Introduction to Hudson or Jenkins, presentation of plug-ins.
- Results reporting.

Hands-on work
Implementation with Hudson SVN and Make. Reporting of coverage and test results, integration with Testlink.

(6) Conclusion

- How to set up automation.
- "Keyword testing": demonstration with robot framework.
- Choice of tools. Open source or paid tools?

Dates and locations

REMOTE CLASS
2026 : 2 June, 13 Oct.