

Course : Open Source, implementing the Software Factory

Practical course - 5d - 35h00 - Ref. OSU

Price : 3540 CHF E.T.

Supporting development teams, a software factory brings together IDE, versioning, testing, deployment and build services. This course presents how to set up such a factory, using the best-known Open Source software in the field: GIT, JUnit, JMeter, Maven, Jenkins...

Teaching objectives

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

- ✓ Understanding the different stages of the software lifecycle
- ✓ Building and deploying a development infrastructure
- ✓ Set up construction tools and integration tests
- ✓ Learn best practices in continuous integration processes
- ✓ Understand collaborative and community work methods, and the main dedicated tools
- ✓ Know how to assess risks and liaise with end-users

Intended audience

Developers, project managers.

Prerequisites

Basic knowledge of software development. Knowledge of the Java EE platform and the Java language is a plus, but not essential.

Practical details

Hands-on work

All the programs created during the lab are available in skeleton form, which participants can complete themselves.

Course schedule

PARTICIPANTS

Developers, project managers.

PREREQUISITES

Basic knowledge of software development. Knowledge of the Java EE platform and the Java language is a plus, but not essential.

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 The fundamentals

- Software lifecycle stages (design, development, acceptance, production).
- The software development model: V-model, incremental and iterative.
- Software factory, continuous integration.
- Different types of environment (development, acceptance, production).
- The plant on a daily basis. The main tools used.

2 Version management

- Features.
- The different source managers: centralized or distributed.
- Change integration issues.
- The role of branches and tags.

Hands-on work

Handling a GIT repository.

3 Build implementation and automation

- Setting up a build server, different tools and types of installation.
- Configuration: main configuration page, GIT/SVN configuration, mail server.
- Jenkins and build, best practices and recommended methods.
- Notification strategies and techniques.
- Fix dependencies between build jobs.
- Jenkins and Maven: reminder of Maven, Maven build configuration, deployment in a Maven repository.

Hands-on work

Create and configure a Maven Jenkins project based on a GIT repository.

4 Test automation

- What tests and what for?
- Test environments.
- Automating unit and integration testing. PHPUnit, JUnit, ...
- Report configuration.
- Measure test coverage.
- Automated acceptance testing.
- Automating performance tests with JMeter.
- Optimize test execution times.

Hands-on work

Adaptation of the Jenkins project to include JUnit and performance test monitoring with JMeter.

5 Validation tests

- Validation test requirements.
- Test repositories and their implementation.
- Test data generators.
- Test robot functionalities.
- HMI test robots.
- Web application test robots.

Hands-on work

Implementation of a test repository, test data generator, GUI test robots, Web application test robots.

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.

6 Automated deployment

- Set up the deployment script.
- Database updates.
- Minimum tests. Back to the past.

Hands-on work

Adaptation of the Jenkins project to automate deployment.

7 Metrics

- Report generation.
- Analysis and reporting tools (PMD, Findbugs, Cobertura, Emma, Checkstyle, Jira Software...).
- Publication of results.

Hands-on work

Setting up metrics tools.

8 Quality

- Reduce risks with continuous integration.
- Understand collaborative teamwork. User contribution to tools.
- Working with end-users.
- Interaction with bug management tools, Bugzilla, Mantis, Bug Tracking...
- Managing an Open Source migration project: market tool study, risks, benefits.

Hands-on work

Anomaly management tools.