

# Formation : Open Source, mise en œuvre de l'Usine Logicielle

*Cours pratique - 5j - 35h00 - Réf. OSU*

*Prix : 3540 CHF H.T.*

Support aux équipes de développement, une usine logicielle rassemble les services : IDE, versioning, tests divers, déploiement, build. Cette formation présente la mise en œuvre d'une telle Usine à l'aide des logiciels Open Source les plus connus du domaine : GIT, JUnit, JMeter, Maven, Jenkins...

## Objectifs pédagogiques

À l'issue de la formation, le participant sera en mesure de :

- ✓ Connaître les différentes étapes du cycle de vie logiciel
- ✓ Construire et déployer une infrastructure développement
- ✓ Mettre en place les outils de construction et les tests d'intégration
- ✓ S'appropriier les bonnes pratiques des process d'intégration continue
- ✓ Appréhender les méthodes de travail collaboratif et communautaire, ainsi que les principaux outils dédiés
- ✓ Savoir évaluer les risques et assurer les liens avec les utilisateurs finaux

## Public concerné

Développeurs, chefs de projet.

## Prérequis

Connaissances de base en développement logiciel. La connaissance de la plateforme Java EE et du langage Java sont un plus, mais non indispensable.

Vérifiez que vous avez les prérequis nécessaires pour profiter pleinement de cette formation en faisant [ce test](#).

## Méthodes et moyens pédagogiques

### Travaux pratiques

Tous les programmes réalisés en TP existent sous forme de squelettes que les participants complètent eux-mêmes.

### PARTICIPANTS

Développeurs, chefs de projet.

### PRÉREQUIS

Connaissances de base en développement logiciel. La connaissance de la plateforme Java EE et du langage Java sont un plus, mais non indispensable.

### COMPÉTENCES DU FORMATEUR

Les experts qui animent la formation sont des spécialistes des matières abordées. Ils ont été validés par nos équipes pédagogiques tant sur le plan des connaissances métiers que sur celui de la pédagogie, et ce pour chaque cours qu'ils enseignent. Ils ont au minimum cinq à dix années d'expérience dans leur domaine et occupent ou ont occupé des postes à responsabilité en entreprise.

### MODALITÉS D'ÉVALUATION

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...

Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

## Modalités d'évaluation

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...

Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

## Programme de la formation

### 1 Les fondamentaux

- Les étapes du cycle de vie du logiciel (Conception, développement, recettage, mise en production).
- Le modèle de développement logiciels : modèle en V, modèle incrémental et itératif.
- Usine logicielle, intégration continue.
- Les différents types d'environnements (développement, recette, production).
- L'usine au quotidien. Les principaux outils utilisés.

### 2 Gestion des versions

- Les fonctionnalités.
- Les différents gestionnaires de sources : centralisé ou distribué.
- Les problématiques d'intégration des changements.
- Le rôle des branches et tags.

#### Travaux pratiques

Manipulation d'un dépôt GIT.

### 3 Mise en place et automatisation du build

- Mise en place d'un serveur de build, les différents outils et types d'installation.
- Configuration : page principale de la configuration, configuration de GIT/SVN, serveur de mail.
- Jenkins et le build, les meilleures pratiques et méthodes recommandées.
- Stratégies et techniques de notification.
- Fixer les dépendances entre les travaux de build.
- Jenkins et Maven : rappel sur Maven, configuration du build Maven, déploiement dans un repository Maven.

#### Travaux pratiques

Création et configuration d'un Projet Maven Jenkins basé sur un référentiel (repository) GIT.

## MOYENS PÉDAGOGIQUES ET TECHNIQUES

- Les moyens pédagogiques et les méthodes d'enseignement utilisés sont principalement : aides audiovisuelles, documentation et support de cours, exercices pratiques d'application et corrigés des exercices pour les formations pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.
- À l'issue de chaque formation ou séminaire, ORSYS fournit aux participants un questionnaire d'évaluation du cours qui est ensuite analysé par nos équipes pédagogiques.
- Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le participant a bien assisté à la totalité de la session.

## MODALITÉS ET DÉLAIS D'ACCÈS

L'inscription doit être finalisée 24 heures avant le début de la formation.

## ACCESSIBILITÉ AUX PERSONNES HANDICAPÉES

Pour toute question ou besoin relatif à l'accessibilité, vous pouvez joindre notre équipe PSH par e-mail à l'adresse [psh-accueil@orsys.fr](mailto:psh-accueil@orsys.fr).

## 4 Automatisation des tests

- Quels tests et pour faire quoi ?
- Les environnements de tests.
- Automatisation des tests unitaires et d'intégration. Les bibliothèques PHPUnit, JUnit, ...
- Configuration des rapports.
- Mesurer la couverture de test.
- Automatisation des tests d'acceptation.
- Automatisation des tests de performance avec JMeter.
- Optimiser les temps d'exécution des tests.

### Travaux pratiques

Adaptation du projet Jenkins pour inclure JUnit et suivi des tests de performance avec JMeter.

## 5 Les tests de validation

- Les besoins en test de validation.
- Les référentiels de test et leur mise en œuvre.
- Les générateurs de données de test.
- Les fonctionnalités des robots de test.
- Les robots de test d'IHM.
- Les robots de test d'applications Web.

### Travaux pratiques

Mise en œuvre d'un référentiel de test, d'un générateur de données de test, de robots de test d'IHM, de robots de test d'applications Web.

## 6 Automatisation du déploiement

- Mise en place du script de déploiement.
- Mise à jour des bases de données.
- Tests minimaux. Retour en arrière.

### Travaux pratiques

Adaptation du projet Jenkins pour automatiser le déploiement.

## 7 Les métriques

- La génération de rapports.
- Les outils d'analyse et de reporting (PMD, Findbugs, Cobertura, Emma, Checkstyle, Jira Software...).
- La publication des résultats.

### Travaux pratiques

Mise en place des outils de métriques.

## 8 Qualité

- Réduire les risques grâce à l'intégration continue.
- Appréhendez le travail collaboratif dans une équipe. La contribution des utilisateurs aux outils.
- Travailler avec les utilisateurs finaux.
- Interaction avec les outils de gestion des anomalies, Bugzilla, Mantis, Bug Tracking...
- Gérer un projet de migration à l'Open Source : étude des outils du marché, risque, avantage.

### Travaux pratiques

Utilisation d'outils de gestion des anomalies.