

## Outils de productivité

### Mise en place d'une usine logicielle java

Cette formation présente les différents outils préliminaires à la mise en place d'une usine logicielle.

Les tests unitaires avec JUnit et fonctionnels avec JMeter, l'automatisation du build et des tests avec Ant et Maven et la gestion de versions avec Subversion y sont abordés comme un prérequis à l'Intégration Continue.

Cette formation est conçue comme la préparation à nos formations Jenkins et à l'Introduction à CloudBees qui achèvent la mise en place de l'usine logiciel par la mise en oeuvre de l'Intégration Continue.

#### Détails

- **Code** : UL-OPR
- **Durée** : 4 jours ( 28 heures )

#### Public

- Architectes
- Chefs de projets
- Développeurs

#### Pré-requis

#### Objectifs

- Comprendre l'IC et ses prérequis
- Mettre en oeuvre les tests (JUnit, JMeter, ...)
- Gérer ses sources avec Subversion
- Automatiser le build avec Ant/Maven

#### Programme

##### Introduction

- Les tests unitaires
- Objectifs
- Définition
- Les concepts du TDD
- L'architecture d'intégration
- L'intégration continue

##### JUnit

- Objectif des tests unitaires
- Les concepts
- Test case
- Assertions
- Tests avec JUnit 4
- Test fixtures
- Pré et post processing
- Test de levée d'exception
- Test de durée
- Assertions spéciales
- Suites de tests
- Lancement des tests
- Les bonnes pratiques
- Intégration dans les IDE
- Les Mock Objects
- Le pattern ObjectMother
- Principe des Mock Object
- Utilisation des Mock Object
- Mock statique / dynamique

- Exemple avec EasyMock
- Exemple avec JMockit

##### ANT

- Philosophie ANT
- Cible / Target
- Tâches ANT
- Gestion des fichiers
- Gestion des archives
- Les ensembles de fichiers FileSet
- Pattern
- Inclusion et Exclusion
- Gestion des commandes
- Exécuter ANT
- Utiliser les Properties
- Commande java
- JUnit avec Ant
- Intégration de ANT dans Eclipse
- Création de tâches ANT personnalisées
- Les bonnes pratiques

##### Subversion

- Objectifs
- La gestion de versions de projets informatiques
- Le problème du partage de fichier
- Objectifs et positionnement par rapport à CVS
- Les fonctionnalités de Subversion
- Les choix d'architecture
- Protocole supportés

- Organisation des révisions
- États possibles d'un fichier
- Installation de Subversion
- Créer le référentiel
- Intégration dans Eclipse
- Organisation du référentiel
- Importer des données dans un référentiel
- Travail sur des fichiers : checkout
- Extraction d'une copie de travail à partir d'un référentiel (checkout)
- Déverser les modifications vers le serveur
- Récupérer les modifications faites par les autres développeurs
- Les patches
- Les conflits
- Les branches de développement
- Les tags
- Opérations utiles

## Maven

- Présentation de Maven
- Maven vs. Ant
- Principes de Maven
- Architecture de Maven
- POM
- Goal
- Cycle de vie du build
- Les repositories
- Recherche de l'artefact
- Les dépendances – portée ou « scope »
- Les propriétés
- Archetype
- Maven et Eclipse
- Les plugins
- Gestion multi-projets
- Multi-projet – pom parent
- Multi-projet – Modules
- Figurer les numéros de version
- Packaging et installation de projet
- Création de site
- Gestion des ressources
- Gestion des dépendances
- Gestion des conflits
- Gestion des versions
- Gestion des profils

- Gestion du déploiement
- Site documentaire pour le projet

## Tests fonctionnels – JMeter

- Les fondamentaux du test logiciel
- Ingénierie des tests logiciel
- Boîte noire / tests fonctionnels
- Boîte blanche / tests structurels
- Tests de non régression et Smoke
- Tests statiques vs tests dynamiques
- Les différents types de tests
- Méthodologie
- Effort de test
- Les principes du TDD
- Agile Model-Driven Development
- Les tests fonctionnels
- Panorama des outils de tests fonctionnels
- Apache JMeter
- Principe général
- Protocoles supportés
- Usages possibles
- Construction d'un scénario de test
- Terminologie
- Simulation d'utilisateurs dans JMeter
- Capture des scénarios
- Développement
- Préparation et conduite des tirs
- Analyse
- Les causes de mauvaises performances
- Possibilités d'optimisation
- Développer un plan de tests
- Plan de test / de travail
- Groupe d'unités
- Catégories des éléments de JMeter
- Échantillons (Samplers)
- Contrôleurs logiques
- Récepteurs (Listeners)
- Compteurs de temps (Timers)
- Assertions
- Éléments de configuration
- Variabilisation des scénarios
- Pré et Post-processeurs
- Pré et Post-processeurs
- Composants autres
- Architecture possible
- Limitations de JMeter

## Modalités

- **Type d'action** :Acquisition des connaissances
- **Moyens de la formation** :Formation présentielle – 1 poste par stagiaire – 1 vidéo projecteur – Support de cours fourni à chaque stagiaire
- **Modalités pédagogiques** :Exposés – Cas pratiques – Synthèse
- **Validation** :Exercices de validation – Attestation de stages