



zenika
ARCHITECTURE INFORMATIQUE

www.zenika.com

Intégration Continue

Un complément à la formation Maven

Contexte

L'intégration continue est un principe majeur d'industrialisation des développements. Il permet de contrôler le code et de le tester de manière automatique tout en fournissant un système d'alertes en cas de problèmes. Le formateur possède une expérience industrielle grand compte de ces problématiques et est commiteur Hudson. Cette formation est le complément idéal à la formation [Maven](#). Pour une formation avancée à l'intégration continue, veuillez consulter la fiche [Hudson](#)

Objectifs

- Comprendre les objectifs et les enjeux de l'intégration continue
- Apprendre les bonnes pratiques

Contenu

Introduction

- Principes et pré-requis
- Types de construction d'une application
 - Construction à la demande
 - Construction automatisée

Le serveur d'intégration Hudson

- Historique
- Comparaison avec les autres outils du marché (CruiseControl, Continuum, Bamboo, TeamCity, ...)
- Architecture et ensemble des plugins offerts

Automatisation de la construction

- Comparaison de l'utilisation des IDE et de l'intérêt d'un moteur d'intégration couplé à un script de construction
- Introduction aux langages de construction : Ant, Maven, Gant, Gradle, ...
- Intérêts et limites des environnements automatisés
- Sensibilisation au temps de construction d'une application
- Validation de la chaîne de construction

Déploiement et gestion des livrables

- Stratégie de mise à disposition du résultat construit
 - Cas d'une application Web
 - Cas d'un web service
 - Cas d'une application RCP
- Traçabilité des livrables
- Notion de dépôt (repository)

Exécution des tests dans le processus de

construction

- Intérêts de l'automatisation des tests
 - Concepts XP « l'extreme Programming »
 - Principes du TDD « Test Driven Development »
- Présentation des familles de tests
 - Tests unitaires
 - Tests d'intégrations
 - Tests fonctionnels
- Présentation des outils de tests de type Xunit (JUnit, Dbunit, TestNG, ...)

La communication avec un gestionnaire de sources

- Les différents types de stratégie
 - Notion de « tronc », « branches » et « étiquettes »
 - Comparaison de la politique de branches par version par rapport à la politique de branches par fonctionnalité

Qualité de code et documentations

- Génération automatique des métriques de codes
 - Amélioration des standards de code et des bonnes pratiques avec Checkstyle et PMD
 - Détection d'erreurs potentielles avec FindBugs
 - Couverture de code avec Emma
 - Génération de la documentation UML avec UMLGraph
 - Génération des schémas de bases de données avec SchemaSpy
- Visualisation à travers le dashboard Hudson

Communication

- Principes
- Indicateurs de visibilité du résultat construit
 - Statut, Santé et tendance
 - Savoir réagir
- Gestion des utilisateurs et communication avec des outils externes
 - Notification par email et flux RSS

Durée

1 jour

Répartition

20% théorie

80% pratique

Pré-requis

Expérience de développement Web Java

Public

Architecte
Développeur
Chef de projet

Tarif (déjeuners inclus)

800 € (HT)

Lieu

Paris 9ème

Sessions 2009

14 au 14 janvier

18 au 18 février

1 au 1 avril

6 au 6 mai

17 au 17 juin

17 au 17 juillet

3 au 3 septembre

21 au 21 octobre

9 au 9 décembre

Intra-entreprise sur demande

Inscription et renseignements

+33(0)1.45.26.19.15

training@zenika.com

www.zenika.com