|  |
| --- |
| **INTEGRATION CONTINUE**  **ET METHODES AGILES** |
| 5 jours |
| **OBJECTIFS :** |
| **Automatisez vos builds et déploiements. Générez des rapports qualitatifs. Auditez le code source. Prévenez les bugs et la régression. Limitez les risques de dérive. Gérez les anomalies.**  **Maîtriser les maillons d’une chaîne d’intégration continue Automatiser la construction de vos applications Tester vos applications en continue Mettre en place un tableau de bord de vos intégrations Générer des rapports liés à l’activité de développement Valider le respect des normes de développement par les développeurs Déclencher des alertes et notifications Prévenir les bugs et la régression de vos applications Automatiser la création de vos livrables** |
| **PARTICIPANTS/PRE-REQUIS** |
| Destiné à des profils expérimentés Java/Jee |
| **PROGRAMME** |
| **Introduction**  Chaine d’intégration continue  les 7 étapes de mise en oeuvre intégration continue  Role du serveur d’Intégration continue  La place des tests unitaires / intégration  Historique Jenkins / Hudson  Lequel choisir et quand : Jenkins OU Hudson ?  **Maven pour de développeur**  définition  Comparaison Ant  conventions  Installation  Coordonnées  sections de pom.xml  CréationProjet maven  plugin, goal, Mojo |
| Phases maven  Repository  plugin m2eclipse  plugin checkstyle |
| **Maven pour le chef de projet**  projet multi-modules  héritage  Composition  génération rapports  plugin jxr  plugin cobertura  plugin changelog  plugin pmd  plugin javadoc  plugin findbugs  plugin Sonar  **Repository d’entreprise**  Définition offre du marché  Focus sur Nexus  Proxy  livraisons  Installation  Intégration  **Serveur intégration continue**  définition  avantages  positionnement  Focus Hudson / Jenkins  Rapports  Build quotidiens  Graphes  Qualité |

|  |
| --- |
| **Premiers pas Jenkins**  Préparer l’environnement  Installation rapide  Configuration rapide  Création premier Job  Lancement premier Job  Rapport Javadoc  Rapport couverture de code  **Installation professionnelle de Jenkins**  Téléchargement et installation  Configurer du serveur  Installer Jenkins sous Linux  Installer Jenkins sous Windows  Installer Jenkins en tant que service  Installer Jenkins derrier Apache Httpd  Mémoire necessaire  répertoire USER\_HOME/.jenkins  Backup données Jenkins  Upgrade Jenkins |
| **Configuration professionnelle de Jenkins**  Tableau de bord de la configuration Jenkins  Configuration Variables d’environnement Système  Configuration Propriétés globales  Configuration outils de build : Maven, Ant, Shell  Configuration gestionnaire de source (Subversion, cvs) Configuration du serveur de mail  Configuration du serveur proxy |

**Les Jobs**

Notion de Job Jenkins

Création d’un job ’freestyle’

Configuration gestionnaire de source

Les triggers les étapes

Actions post-build

Lancement de build MAven

Voir les résultats d’un job

**Automatiser les tests**

*Automatiser tests unitaires*

Automatiser tests d’intégration

Configurer les rapports de test dans Jenkins

Afficher les résultats de test dans

Jenkins Ignorer les tests

Couverture de code

Automatiser tests d’acceptance

Automatiser tests de performance avec JMeter

Résoudre les problèmes de lenteur des tests

**Sécurité**

Authentification & Autorisation

Activer la sécurité Jenkins

Sécurité simple

Realms : royaumes de de sécurités

Audit de sécurité Jenkins

**Notification**

Introduction

Notification mail

Notification RSS

Notification IRC

Notification SMS

Notification bruit

Notification pda

**Qualité du code**

La qualité de code dans le processus de build

Outils populaire : PMD, Checkstyle, FindBugs, CodeNarc

Rapport outils qualité de code : plugin ’Violations’

Rapport sur la complexité du code

Rapport sur les taches ouvertes (FIXME, TODO, )

Intégration avec l’outil Sonar