

Formation Intégration Continue

Industrialisation du développement

Référence : IC-03

Durée : 3 jour(s)

Présentation

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/Jenkins.

Objectifs

- Comprendre les fondamentaux et les enjeux de la mise en place de l'intégration au sein de ses projets afin d'améliorer la communication et la collaboration entre les différentes équipes
- Savoir gérer les dépendances de construction de ses composants
- Comprendre et résoudre les enjeux de traçabilité des objets construits
- Connaître les principes et les bonnes pratiques dans la mise en place d'outils de métriques
- Maîtriser l'utilisation et la mise en place du serveur d'intégration continue Hudson/Jenkins

Répartition: 50% Théorie, 50% Pratique

Public: Architecte, Leader Technique, Développeur, Chef de projet, Responsable qualité

Pré-requis: Connaissance de base en développement logiciel

Programme

Comprendre les enjeux de l'intégration continue

- Première approche et notion de processus d'intégration
- Les apports de l'intégration continue et les risques réduits dans les équipes de développement
- De quoi a-t-on réellement besoin pour commencer une démarche d'intégration continue?

Changement, déclenchement et processus d'intégration

- Notion d'événement
- Détail des différentes étapes d'un processus d'intégration
- Les types de déclenchement du processus

Automatisation de la construction

- Les enjeux
- Introduction aux outils de build Java : Ant/Ivy, Maven, Gradle, ...
- Comparaison de l'utilisation des IDE et de l'intérêt d'un moteur d'intégration couplé à un script de construction
- Comment démarrer son automatisation?
- Sensibilisation au temps de construction globale d'une application
- Validation et traçabilité de la chaîne de construction
- Problématique de la reproductibilité

La gestion des composants de son application

- Problématique du découpage en composants de son application
- Présentation des outils de gestion de dépendances
- Les stratégies de l'enchaînement de la construction des composants

Communication avec un gestionnaire de sources

- Les différents types de stratégie en fonction de l'outil SCM
- Politique de branches par version vs politique de branches par fonctionnalité
- Problématique de traçabilité
- Exemple de mise en oeuvre avec le triplet Maven, Hudson/Jenkins et SVN
- Exemple de mise en oeuvre avec le triplet Maven, Hudson/Jenkins et GIT
- Etude du cas particulier : le mécanisme de livraison (release) de Maven

Le serveur d'intégration continue Hudson/Jenkins

- Historique et positionnement vis à vis de la concurrence
- Panorama des fonctionnalités
- Les plugins les plus utilisés
- Installation, configuration et les pièges à éviter
- Première mise en oeuvre avec un projet Maven et un projet Ant/Ivy

Gestion des livrables

- Stratégie de mise à disposition du résultat construit
- Notion de dépôt d'artefacts
- Spectre des principaux outils de gestion de dépôt d'artefacts: Nexus, Artifactory et Archiva

- Problématique de traçabilité du résultat et des informations de build
- Panorama des plugins de gestion des livrables dans Hudson/Jenkins

Le déploiement

- Pourquoi déployer?
- Présentation des différents cas d'utilisation
- Jusqu'où aller?
- Les pièges à éviter
- Exemple de mise en oeuvre avec une application Web et une application RCP orchestré par Hudson/Jenkins

Les tests dans le processus de construction

- Intérêts de l'automatisation des tests
- Concepts de « l'extreme Programming »
- Principes des « TestDriven Development »
- Présentations des familles de tests
- Tests unitaires, tests d'intégrations et tests fonctionnels
- Présentation des outils de tests de type Xunit (JUnit, Dbunit, TestNG, ...)
- Apport du langage Groovy pour l'écriture des tests
- Exemple de test d'une application Web avec Selenium
- Exemple de test d'un Web Service avec JMeter et SOAPUI
- Exemple de test d'une application RCP avec SWTBot

Qualité de code et traitement des résultats

- Apports de la mise en place d'outils de métriques
- Les principaux outils de métriques Java: de la génération de la documentation à la couverture de code en passant par la détection de bugs
- Exemple du site Maven
- Installation et configuration de Sonar
- Mise en oeuvre d'une qualimétrie avec Sonar pour un projet Maven et un projet Ant/Ivy
- Publication des résultats dans le serveur Hudson/Jenkins à travers ses plugins
- Les bonnes pratiques d'utilisation
- Les solutions pour les autres langages que Java dans Hudson/Jenkins et Sonar

Communication

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

Gestion de plusieurs configurations

- Problématiques de multi-cibles
- Mise en oeuvre avec Hudson/Jenkins

OFFERT EN INTER-ENTREPRISE

- ☎ Le petit déjeuner croissants, jus d'orange, café)
- ☎ Le déjeuner
- ☎ Une qualification téléphonique si nécessaire avec l'un de nos consultants

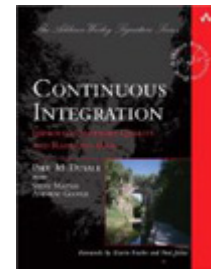
Tel: +33(0)1 45 26 19 15
Fax : +33(0)1 75 43 49 92
Email : training@zenika.com

Auteur du cours



Grégory Boissinot est un spécialiste des outils de build et des problématiques d'intégration continue qu'il a mis en place à grande échelle. Grégory possède une expérience industrielle grand compte de ces problématiques et contribue activement à la communauté **Open Source** dans ce domaine en étant l'un des principaux commiteurs Hudson/Jenkins mondiaux.

Livre offert ! (*)



Continuous Integration: Improving Software Quality And Reducing Risk

(*) Les livres sont offerts uniquement pour les formations inter-entreprise. Zenika se réserve le droit de changer le livre proposé à tout moment.