INTÉGRATION CONTINUE





INTRODUCTION



REGLE 1 : PLUS IL Y A DU CODE, PLUS IL Y A D'ERREURS



REGLE 2: PLUS ON TEST, MOINS ON CORRIGE





REGLE 3 : LA MACHINE EST PLUS FIABLE QUE L'HOMME



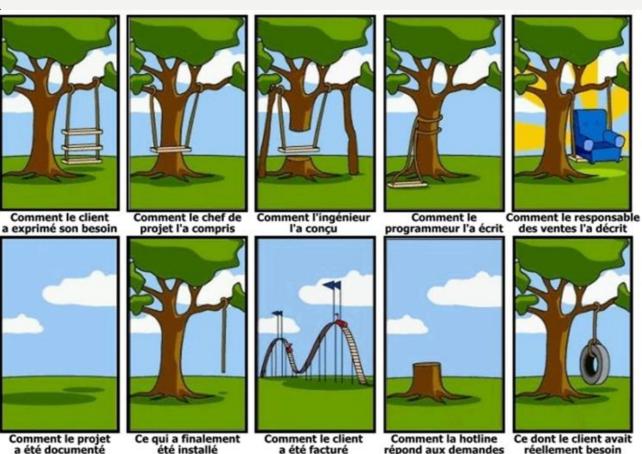


REGLE 4: PLUS ON DIVISE, MIEUX ON COMPREND



REGLE 5: PLUS ON LIVRE, MOINS ON RISQUE LES

ECARTS





LA SOLUTION

L'intégration continue est un ensemble de pratiques utilisées en <u>génie logiciel</u> consistant à vérifier à chaque modification de <u>code source</u> que le résultat ne produit <u>pas de régression</u> dans l'application.

L'INTEGRATION CONTINUE

Les composants essentiels











L'INTEGRATION CONTINUE

Glossaire

| Sigle | Signification | Exemples d'outils |
|-------|---------------------------------------|---|
| RM | Repository Manager | Nexus, Apache Archiva, Artifactory |
| Cl | Continuous Integration | Jenkins/Hudson, TeamCity, Bamboo, Continuum, GitLab |
| QA | Quality Assurance | SonarQube |
| SCM | Source Control Management | CVS, SVN, GIT , Bazaar, Mercurial |
| IDE | Integrated Development Environment | NetBeans, Eclipse, IntelliJ |

L'INTEGRATION CONTINUE

Les principes de base

- Aider les développeurs dans leurs tâches
- Aider les équipes dans le déploiement
- Faciliter les livraisons
- Garantir la qualité de l'application
- Eviter les erreurs humaines
- Réduire les tâches récurrentes à faible valeur ajouté



GENESE DU CODE



Le poste du développeur dans l'intégration continue











L'IDE

Choisir le bon IDE:

- Performance
- Prix
- Support/communauté
- Richesse des plugins
- •











Le gestionnaire de construction projet

- Structurer le code
- Gérer le cycle de vie de construction projet
- Gérer les dépendances
- Automatiser des tâches sur le code
- Générer la documentation
- ...









Maven: un outil de construction pour JAVA

- Outil de construction de projets JAVA J2EE
- Projet Open Source de la fondation Apache
- Projet initié en 2001 (parmi d'autre projets similaires)
- Successeur du projet ANT

Maven: un outil de construction pour JAVA

• Convention plutôt que configuration

| Répertoire | Contenu | |
|----------------------|--|--|
| /src | les sources du projet (répertoire qui doit être ajouté dans le gestionnaire de sources) | |
| /src/main | les fichiers sources principaux | |
| /src/main/java | le code source (sera compilé dans /target/classses) | |
| /src/main/resources | les fichiers de ressources (fichiers de configuration, images,). Le contenu de ce répertoire est copié dans target/classes pour être inclus dans l'artéfact généré | |
| /src/main/webapp | les fichiers de la webapp | |
| /src/test | les fichiers pour les tests | |
| /src/test/java | le code source des tests (sera compilé dans /target/test-classses) | |
| /src/test/resources | les fichiers de ressources pour les tests | |
| /target | les fichiers générés pour les artéfacts et les tests (ce répertoire ne doit pas être inclus dans le gestionnaire de sources) | |
| /target/classes | les classes compilées | |
| /target/test-classes | les classes compilées des tests unitaires | |
| /target/site | site web contenant les rapports générés et des informations sur le projet | |
| /pom.xml | le fichier POM de description du projet | |

Maven : le cycle de vie standard

Validate

Vérifie que le projet est valide

Installe les artefacts dans le dépôt local

Install

Test

Exécute les tests unitaires

Génère les artefacts

Package

Compile

Compile le projet

Maven : les nivaux de configuration

global
 Configuration générale implicite utilisée par Maven
 Settings.xml
 Configuration locale d'environnement
 Configuration projet



EXERCICE PRATIQUE



- Installer maven
- Créer un projet java Maven helloworld
- Compiler le projet avec maven (mvn compile)
- Ecrire un test unitaire

20

- Lancer les tests unitaires avec maven (mvn test)
- Créer un artefact avec maven (mvn package)

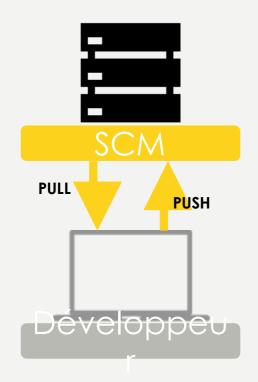




LE SCM

VERSIONNER SON CODE

Le SCM dans l'intégration continue









Pourquoi faire?

- Entrepôt du code source
- Versionner son code
- Travailler en équipe sans se marcher dessus
- Stocker le code de manière pérenne
- Distribuer le code de manière homogène et sécurisée
- Partager le code de façon sécurisée







GIT : Le SCM par excellence

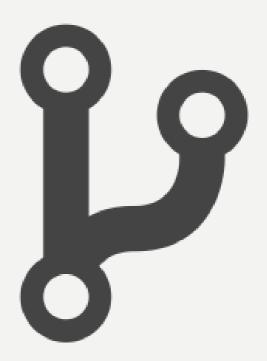
- SCM Open Source décentralisé créé par Linus Torvalds en 2005 initialement pour des besoins personnels
- Conçu et optimisé pour le noyau UNIX et ultérieurement porté sur les autres plateforme (dont WINDOWS et MAC)
- Depuis 2016 le SCM le plus populaire au sein de la communauté des développeurs, notamment grâce à son mode décentralisé
- Actuellement rendu à sa version 2.20 (datant de décembre 2018)

GIT: Les avantages

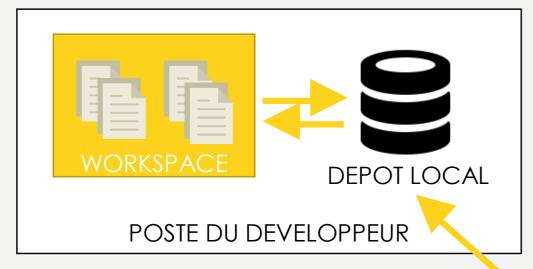
- Permet de travailler aisément sur de nombreuses branches en parallèle
- Compatible avec de nombreux protocoles d'échanges de fichiers (FTP, HTTP, SFTP, HTTPS, SSH, etc...)
- Adapté aux projets de taille modeste ou de taille importante
- L'ensemble de l'historique est chiffré
- De nombreux algorithmes de résolution de conflits intégrables
- Haute performance

GIT : Choisir la bonne méthodologie

- Taille de l'équipe projet
- Répartition géographique des membres du projet
- Politique de release (fréquence)
- Intégration continue ?
- Complexité du workflow
- Besoins de rigidité, ou au contraire de souplesse



GIT: Architecture



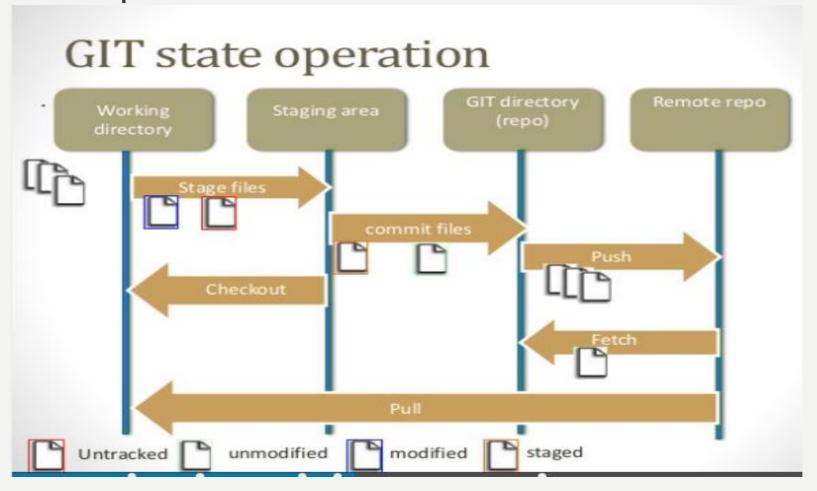




12/01/2021 27



GIT : Les étapes



12/01/2021

28



EXERCICE PRATIQUE



- Installer GIT
- Créer un dépôt git local sur le projet helloworld
- Créer un dépôt distant github
- Envoyer le code source sur le dépôt git (push)



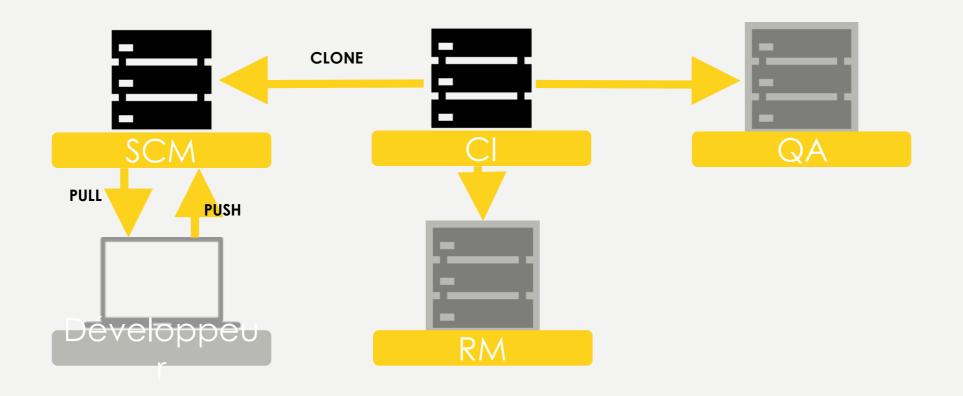
12/01/2021 29



LE SERVEUR D'INTEGRATION CONTINUE

AUTOMATISER LES TACHES

Le serveur CI dans l'intégration continue



Quelques outils du marché









Jenkins: un serveur CI (très) populaire



- Projet open source issue du fork du projet Hudson
- Une très large communauté
- De nombreux plugins
- Une interface simple et intuitive
- Une approche configuration as code (Pipelines)

Le workflow standard



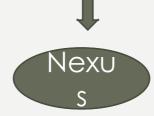
Récupérati on des sources

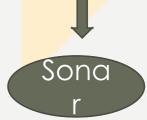
Compilati on

Tests

Génératio n des artefacts

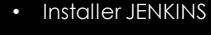
Déploiem ent Génératio n des rapports Publicatio n des rapports











- Créer un JOB de type pipeline
- Le brancher sur le dépôt git distant helloworld
- Créer un fichier Jenkinsfile à la racine du projet helloworld
- Créer dans ce fichier un pipeline qui :
 - Compile le projet
 - Lance les tests unitaires
 - Génère un artefact
 - Le stock sur le dépôt maven local (.m2)
 - Génère un rapport SONAR
 - Affiche un rapport des tests
 unitair



12/01/2021 35



MERCI POUR VOTRE ATTENTION

saadi.abdelmajid@gmail.com

12/01/2021 36