

# L'usine logicielle (ou forge)



**Pour l'industrie du logiciel**

XH

1

## Définition de l'usine logicielle

- Une forge
- Application collaborative de développement logiciel
  - CDE: Collaborative Development Environment
- Cadre commun et standardisé de développement
- Management et Méthodes

XH



## Les services d'une Forge

- Gestion de version
- Système de suivi des incidents
- Un outils de build
- Un outils de partage et de gestion de projet
- Des outils de qualimétrie du code produit et de génération d'états
- Un moteur d'Intégration Continue
  - CI (Continuous Integration )
- Wiki, forum, brèves ...

XH



## Principes de la forge

- Améliorer la qualité des livrables
  - Meilleure maintenabilité du logiciel
  - Traçabilité des changements et des déploiements
  - Améliorer la productivité de l'équipe du projet
- Repose sur des pratiques « agiles » du développement (XP, ...)
  - Fabriquer souvent
  - Tester souvent
  - Intégrer souvent

XH



## Autres points

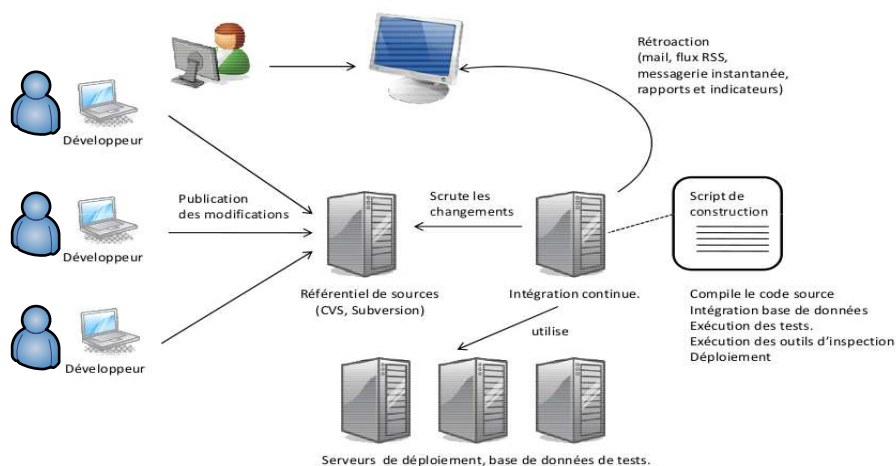
- La forge permet un développement maîtrisé
  - les projets sont construits avec des scripts automatisés
  - la construction des projets est indépendante de l'IDE
- Le développeur exécute une construction privée avant de soumettre son job au CI Hudson
- Elle permet d'assurer le suivi en temps réel des développements selon des critères de qualité

XH

5



## Principes de l'intégration Continue

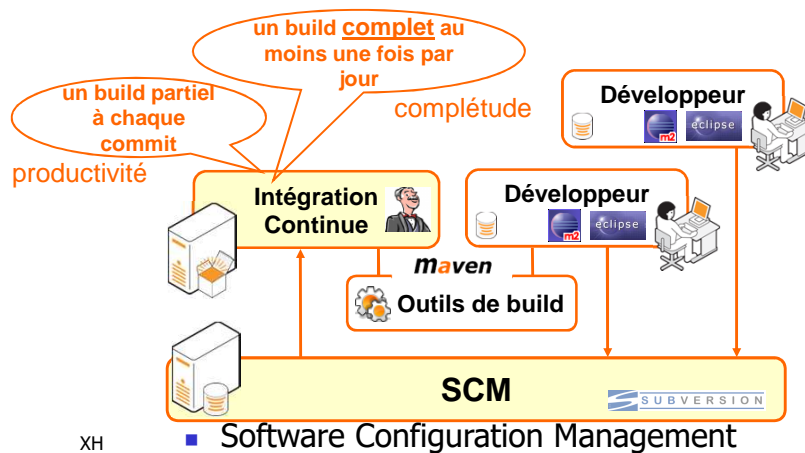


XH

6

## Usine Logicielle : la productivité

- Détecter et corriger les bugs au plus tôt
- S'assurer que l'intégration de code « nouveau » ne casse pas le composant logiciel



## Utilité d'une forge pendant votre stage? Quelques bonnes raisons

- Il est difficile de suivre la qualité d'un projet sans métriques de suivi
- Le formateur pourrait disposer de tableaux de bord et d'indicateurs permettant d'évaluer au fil de l'eau la progression du stagiaire
- Le stagiaire disposerait d'une infrastructure avec des bonnes méthodes (best practises) et des outils qui vont lui permettre de juger de sa progression selon des standards de développement établis
  - Le stagiaire est peu sensibilisé à la problématique des tests
- **Préparer le stagiaire aux standards de l'industrie du logiciel**

XH

8



## Exemples d'outils open-source composant une Forge

Gestion des versions des sources avec **Subversion**  
Gestion du cycle de vie des projets avec **Maven**  
Gestion de l'Intégration Continue avec **Hudson**  
Qualimétrie du projet avec le plug-in **Sonar**

XH

9



## Subversion



- Gestion de version
- SCM très populaire
  - Software Configuration Management
- Permet à plusieurs développeurs de travailler de manière collaborative au même code source et gérer les changements effectués dans l'équipe
- Techniques de merge, ou fusion de code, notion de branche, gestionnaire de conflits ...
- Notion de dépôt (repository)

XH

10



## Subversion (suite)

- Opérations permises:
  - checkout/import/update/commit
- Subversion stocke les données dans un repository central
- Utilisé et configuré avec Apache2 sous Linux (Ubuntu)
- <http://svn.apache.org/>

XH

11



## Maven



- Gestion des builds et suivi du cycle de vie des projets avec Maven
- Maven
- Gestion du build de projet
  - comme Ant ?
- Mais déclaratif:
  - ce qu'on veut faire plutôt qu'écriture d'un build.xml

XH

12

The screenshot shows the Wikipedia page for Apache Maven. The browser's address bar displays the URL `https://fr.wikipedia.org/wiki/Apache_Maven`. The page features the standard Wikipedia layout with a sidebar on the left containing navigation links like 'Accueil', 'Portails thématiques', and 'Contribuer'. The main content area has a tabbed interface with 'Article' selected. The title 'Apache Maven' is prominently displayed, followed by a note about homonyms. The article text describes Maven as a tool for managing and automating the production of Java projects, comparing it to Make and Ant. A summary box on the right lists key details: 'Fondateur' (Sonatype), 'Développeur' (Apache Software Foundation), 'Dernière version' (3.3.9), 'Écrit en' (Java), 'Environnement' (Multi-plateforme), 'Type' (Moteur de production), 'Licence' (Apache 2.0), and 'Site web' (maven.apache.org).

WIKIPÉDIA  
L'encyclopédie libre

Article Discussion Lire Modifier Modifier le code Historique Rechercher

## Apache Maven

*Pour les articles homonymes, voir Maven.*

**Apache Maven** est un outil pour la gestion et l'automatisation de production des projets logiciels Java en général et Java EE en particulier. L'objectif recherché est comparable au système Make sous Unix : produire un logiciel à partir de ses sources, en optimisant les tâches réalisées à cette fin et en garantissant le bon ordre de fabrication.

Il est semblable à l'outil Ant, mais fournit des moyens de configuration plus simples, eux aussi basés sur le format XML. *Maven* est géré par l'organisation *Apache Software Foundation*. Précédemment *Maven* était une branche de l'organisation *Jakarta Project*.

*Maven* utilise un paradigme connu sous le nom de *Project Object Model* (POM) afin de décrire un projet logiciel, ses dépendances avec des modules externes et l'ordre à suivre pour sa

Apache Maven	
<b>maven</b>	
<b>Fondateur</b>	Sonatype (d)
<b>Développeur</b>	Apache Software Foundation
<b>Dernière version</b>	3.3.9 (18 novembre 2015) <sup>1</sup>
<b>Écrit en</b>	Java
<b>Environnement</b>	Multi-plateforme
<b>Type</b>	Moteur de production
<b>Licence</b>	Apache 2.0 licence
<b>Site web</b>	maven.apache.org
<a href="#">modifier</a>	

The screenshot shows the website of Objis, a company specializing in Java training. The header includes the company logo and contact information. A navigation bar lists various services and resources. The main content area features a tutorial titled 'Tutoriel Maven N°5 : intégration Maven / eclipse'. The text describes the tutorial's focus on integrating Maven with Eclipse using the m2eclipse plugin. Below the text, there are sections for 'Liens utiles', 'Prérequis, outils et versions', 'Objectifs', and 'Programme'.

Objis  
Nous allons vous faire aimer Java

Formation informatique JAVA  
Web Mobile depuis 10 ans

contact@objis.fr  
Tél : +33 2 37 77 77 77

Clients | 200 tutos | Catalogue | Promotions | #10000Codeurs | Afrique | Contact

### Tutoriel Maven N°5 : intégration Maven / eclipse

Avec Objis, spécialiste formation MAVEN (3j) et formation intégration continue (5j), maîtrisez l'intégration de Maven dans Eclipse en utilisant le plugin m2eclipse. Comprenez le rôle des plugins Subclipse et m2eclipse dans cette intégration.

Ce tutoriel est un extrait de séances pratiques de la formation MAVEN dispensée par Objis.

**Liens utiles**

- Tutoriaux Subversion Objis

**Prérequis, outils et versions**

- Tutoriel Maven N°1 : Installation Maven

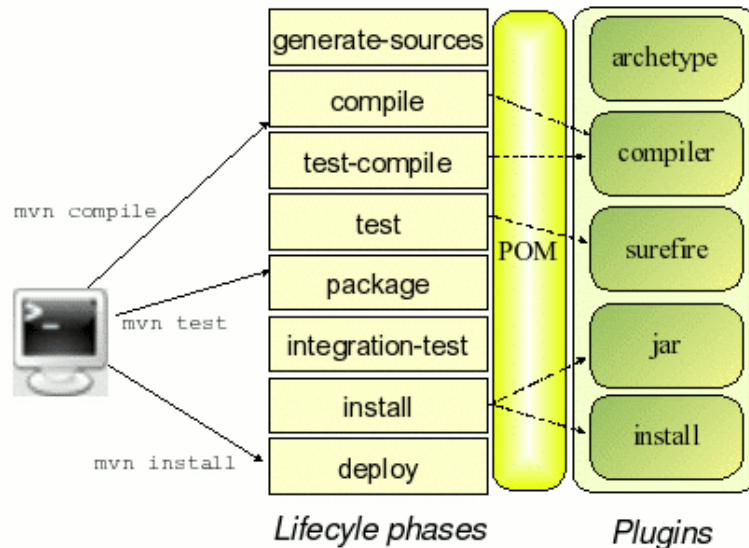
**Objectifs**

- Intégrer Maven et Subversion dans Eclipse

**Programme**



## Les phases du build pour maven



15



## Maven 2 : Les plugins

- |                      |              |                        |                   |
|----------------------|--------------|------------------------|-------------------|
| • Antlr              | • Deploy     | • Javancss             | • Release         |
| • Ant                | • Ear        | • Jboss                | • Repository      |
| • AntRun             | • Eclipse    | • Jcoverage Jdepend    | • Resources       |
| • AspectJ            | • Ejb        | • Jdiff                | • Repository      |
| • Assembly           | • Ejb3       | • Jelly                | • Sablecc         |
| • Assembly-report    | • Exec       | • Jetty                | • Site            |
| • Cargo              | • Groovy     | • Jpox                 | • Slimdog         |
| • Castor             | • Help       | • Jspc                 | • Source          |
| • Changelog          | • Hibernate2 | • Jxr                  | • Surefire        |
| • Changes            | • Idea       | • MAnt                 | • Surefire-report |
| • Commons-attributes | • Install    | • Native               | • Taglist         |
| • Checkstyle         | • Issue      | • One                  | • Tomcat          |
| • Clean              | • It         | • Par                  | • Verifier        |
| • Clover             | • Jalopy     | • Plugin               | • Xslt            |
| • Csharp             | • Jar        | • Pmd                  | • War             |
| • Cobertura          | • Javacc     | • Project-info-reports | • Wsd2java        |
| • Compiler           | • Javadoc    | • Rar                  | • Xdoclet         |
|                      |              |                        | • Xmlbeans        |

Status: [docs.codehaus.org/display/MAVEN/Maven+Plugin+Matrix](http://docs.codehaus.org/display/MAVEN/Maven+Plugin+Matrix)

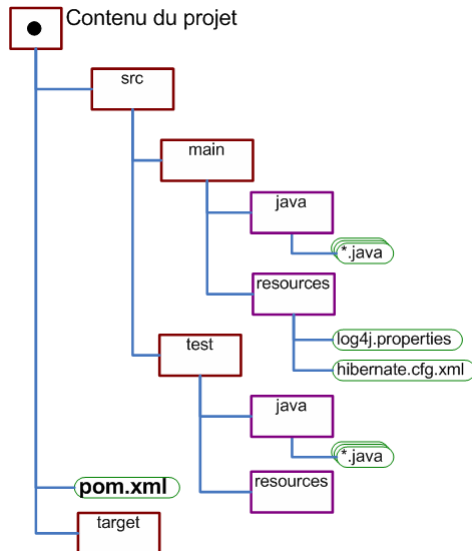
[www.javapolis.com](http://www.javapolis.com)





## Le répertoire d'un projet mavenisé

- Structure du projet commune à tous les IDE (Eclipse, NetBean, ...)

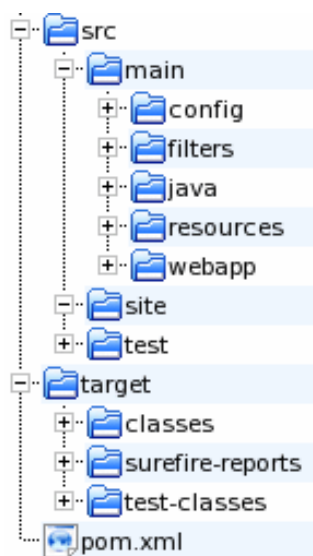


XH

17



## Le répertoire d'un projet



XH

18



## Le pom.xml

- Le fichier du projet pom.xml
- Référentiel des artefacts :
  - définition par groupId / artifactId / packaging / version
- Résolution des dépendances transitives
  - élément <dependencies> du POM
  - Scopes possibles :
    - compile, test, runtime, provided
- ajout de plug-in

XH



## Exemple de pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.mycompany.app</groupId>
  <artifactId>my-app</artifactId>
  <packaging>jar</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>my-app</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

XH



# Hudson (ou Jenkins)



- Gestion de l'intégration continue avec Hudson
- Logiciel d'intégration continue
  - CI (Continuous Integration)
- permet d'assurer le suivi en temps réel des développements selon des critères de qualité
- Hudson est un projet open-source hébergé sur java.net
- Logiciels concurrents:
  - « CruiseControl »
  - « Apache Continuum »

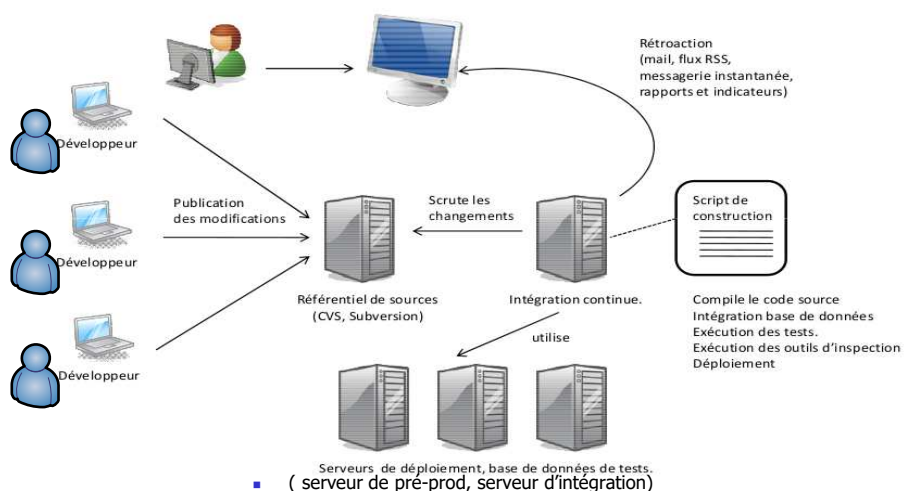


XH

21



## Principes de l'intégration Continue



XH

22

## Tableau de bord Hudson (Jenkins)

The screenshot shows the Hudson dashboard with the following elements:

- Header:** "JBoss QA" and "Hudson".
- Navigation:** "Personnes", "Historique des builds", "Leader board", "Fichier d'attente des builds".
- Jobs List:**

S	W	Job
Blue	Yellow	Arquillian
Blue	Yellow	cdi-tck-nightly
Blue	Yellow	cdi-tck-nightly-1.0
Blue	Yellow	cms-portal-2.6
Blue	Yellow	Concurrent
Blue	Yellow	DB MATRIX JBoss Identity IDM trunk
- Job Details (cdi-tck-nightly-1.0):**
  - Description:** Coverage: Classes 3090/6811 (45%), Methods 13651/44318 (31%), Blocks 295101/1015676 (29%), Lines 68354/233864 (29%).
  - Résultats des tests:** 305 tests en échec sur un total de 1,945 tests.
  - Stabilité du build:** Aucun build récent n'a échoué.

XH

23

## Back office Hudson

The screenshot shows the Hudson 'Back office' with the following elements:

- Header:** "Hudson" and "ENABLE AUTO REFRESH".
- Navigation:** "New Job", "Configure", "Reload Config".
- Build Queue:**

No.	Status
1	Idle
2	Idle
3	Building javanet-maven-repository-daemon #826
4	Building jaxb-ri #3181
5	Building glassfish #105
6	Idle
- Jobs Table:**

Job	Last Success	Last Failure	Last Duration
Common annotations	4 days (#16)	9 months (#3)	39 seconds
bsh	6 months (#11)	10 months (#2)	59 seconds
gdb-parser	6 months (#8)	N/A	1 minute
fi	28 days (#536)	1 month (#567)	7 minutes
fi (weekly)	6 days (#53)	13 days (#52)	5 minutes
glassfish	4 hours (#104)	1 day (#88)	1 hour
hudson	4 minutes (#201)	N/A	1 minute
istack-commons	12 days (#19)	16 days (#5)	14 seconds
japex	3 days (#55)	9 hours (#64)	1 minute
java-ws-xml community discussion updater	4 minutes (#16146)	10 hours (#16125)	1 minute
java.net.aci.processor	18 hours (#162)	N/A	0 seconds

XH

24



## Ce que fait Hudson

- Supporte les tâches ANT
- Supporte les tâches Maven
- Supporte Subversion (SVN)
- Intégration avec:
  - RSS
  - E-mail
  - IM
- Couvre les tests unitaires (JUnit/TestNG)
- Génère la Javadoc

XH

25



## Ce que fait Hudson ... de plus

- En plus de fournir la plupart des fonctionnalités des autres, Hudson offre:
  - Une interface d'administration plus que conviviale
  - Extensible: système de plugins
    - ex: sonar
  - Des builds distribués

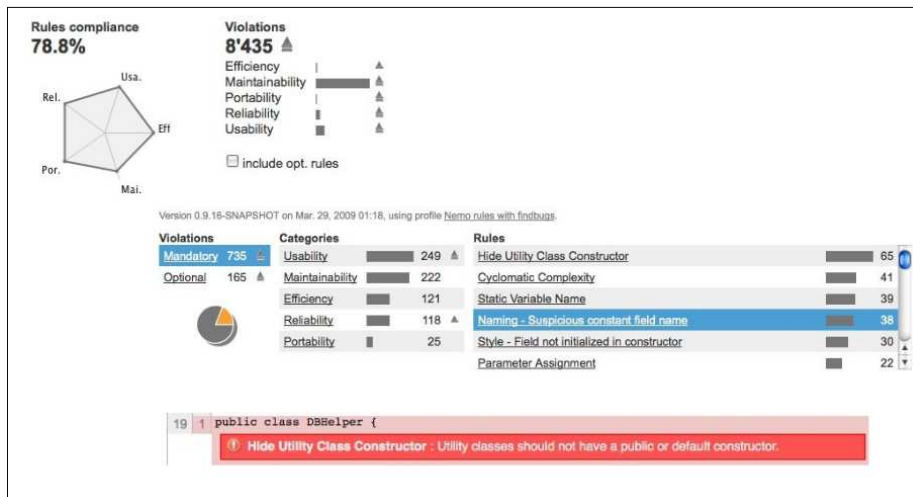
XH

26



## Le plugin Hudson « Sonar »

- Génération de tableaux de bord
- Ici PMD et Checkstyle:



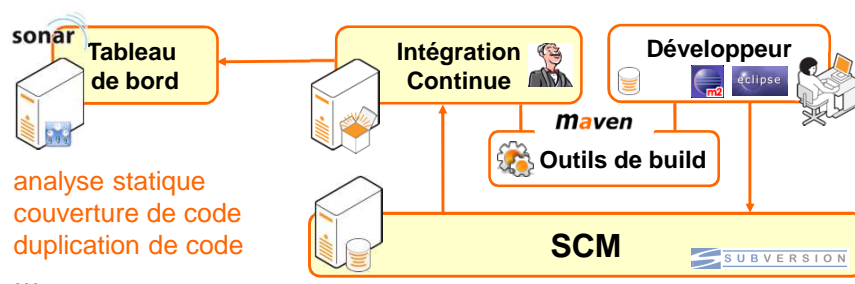
## Qualimétrie du projet avec Sonar

- Qualité du code, respect des conventions avec :
  - Checkstyle ou PMD



## Tableau de bord

- Voir en « temps réel » l'état du projet
- En produisant des rapports régulièrement toute l'équipe peut suivre l'état d'avancement du projet



XH

29



## CheckStyle

- Similaire à PMD
- Javadoc Comments, Naming Conventions, Headers, Imports, Size Violations, Whitespace, Modifiers, Blocks, Coding Problems, Class Design, Duplicate Code

	Description	Resource	In Folder	Location
✗	Name 'rules.FieldNames' must match pattern <code>^[a-z]+(\.[a-zA-Z_][a-zA-Z...]</code>	FieldNames.java	JavaTest/testdata/rules/Fi...	line 2
!	Missing a Javadoc comment.	FieldNames.java	JavaTest/testdata/rules/Fi...	line 5
!	'{' should be on the previous line.	FieldNames.java	JavaTest/testdata/rules/Fi...	line 6
!	Missing a Javadoc comment.	FieldNames.java	JavaTest/testdata/rules/Fi...	line 7
!	Missing a Javadoc comment.	FieldNames.java	JavaTest/testdata/rules/Fi...	line 8
✗	Name 'CONSTANT2' must match pattern <code>^[A-Z]([A-Z0-9]+)*\$</code> .	FieldNames.java	JavaTest/testdata/rules/Fi...	line 8
!	Missing a Javadoc comment.	FieldNames.java	JavaTest/testdata/rules/Fi...	line 10
!	Name 'NON_CONSTANT1' must match pattern <code>^[a-z][a-zA-Z0-9]*\$</code> .	FieldNames.java	JavaTest/testdata/rules/Fi...	line 10
!	Variable 'NON_CONSTANT1' must be private and have accessor methods.	FieldNames.java	JavaTest/testdata/rules/Fi...	line 10
!	Missing a Javadoc comment.	FieldNames.java	JavaTest/testdata/rules/Fi...	line 11
!	Variable 'NON_CONSTANT1' must be private and have accessor methods.	FieldNames.java	JavaTest/testdata/rules/Fi...	line 11



## JDepend

- Calcule les métriques pour les packages java
- Calcul des métriques
- **CC** - Concrete Class Count
  - Nombre de classes concrètes dans le package.
- **AC** - Abstract Class Count
  - Nombre de classes abstraites ou d'interfaces dans le package



## PMD



- Pour du code Java
- Similaire à CheckStyle
- Vérifie
  - variables locales non utilisées/ paramètres / méthodes privées
  - **Bloc catch vide**
  - Bloc instruction 'if' vide
  - Directive d'import dupliquée
  - Classes pouvant être des singletons
  - ...