

# Packaging, automatisation et intégration continue

Licence Professionnelle IDSE  
2015-2016

IUT de Nice

Simon Urli  
[urli@i3s.unice.fr](mailto:urli@i3s.unice.fr)

Avec des morceaux choisis de Sébastien Mosser et Philippe Collet  
(enfin, de leurs cours...)

# Processus de développement

- Développement des fonctionnalités et des tests
- Lancement des tests
- Développement ...
- Lancement ...
- Livraison et/ou déploiement d'une version stable

# Processus de développement

- Développement des fonctionnalités et des tests : **code**
- Lancement des tests : **commandes**
- Développement ... : **code**
- Lancement ... : **commandes**
- Livraison et/ou déploiement d'une version stable : **commandes**

# Processus de développement

- Développement des fonctionnalités et des tests : **code**
- Lancement des tests : **commandes => Script ?**
- Développement ... : **code**
- Lancement ... : **commandes => Script ?**
- Livraison et/ou déploiement d'une version stable : **commandes => Script ?**

# Compiling Code During Lab Sessions



```
azrael:labs mosser$ ls  
Exercice.java Main.java
```

```
azrael:labs mosser$ javac *.java
```

```
azrael:labs mosser$ java Main  
42
```

**Legendary**, isn't it?

2

# Code in Real-Life™



## Languages

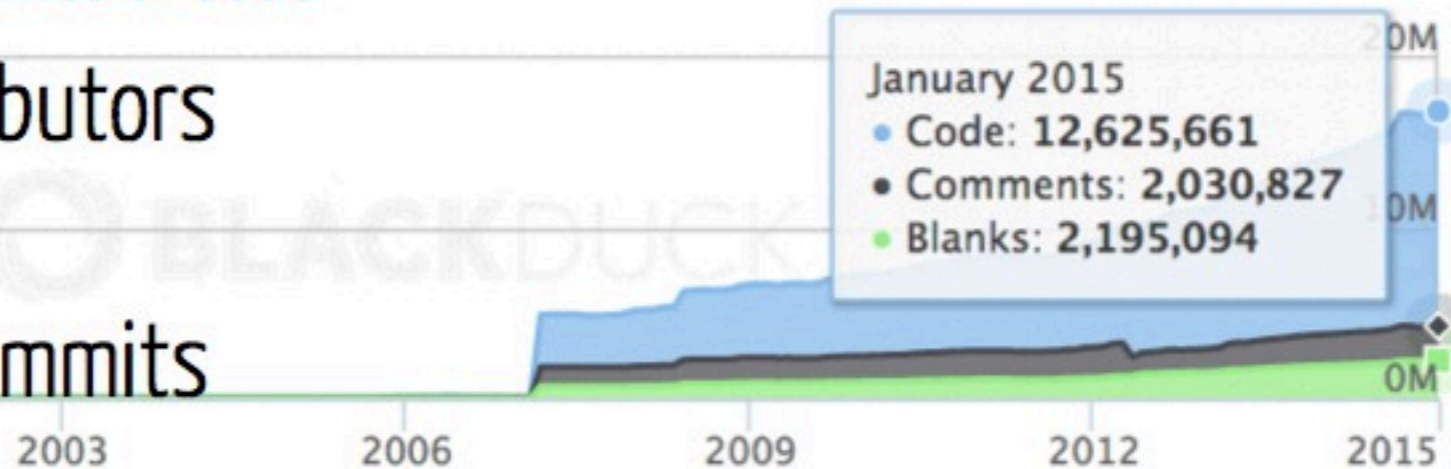


C++	38%	JavaScript	20%
C	17%	31 Other	25%

## Lines of Code

**3,254** contributors

**230,442** commits

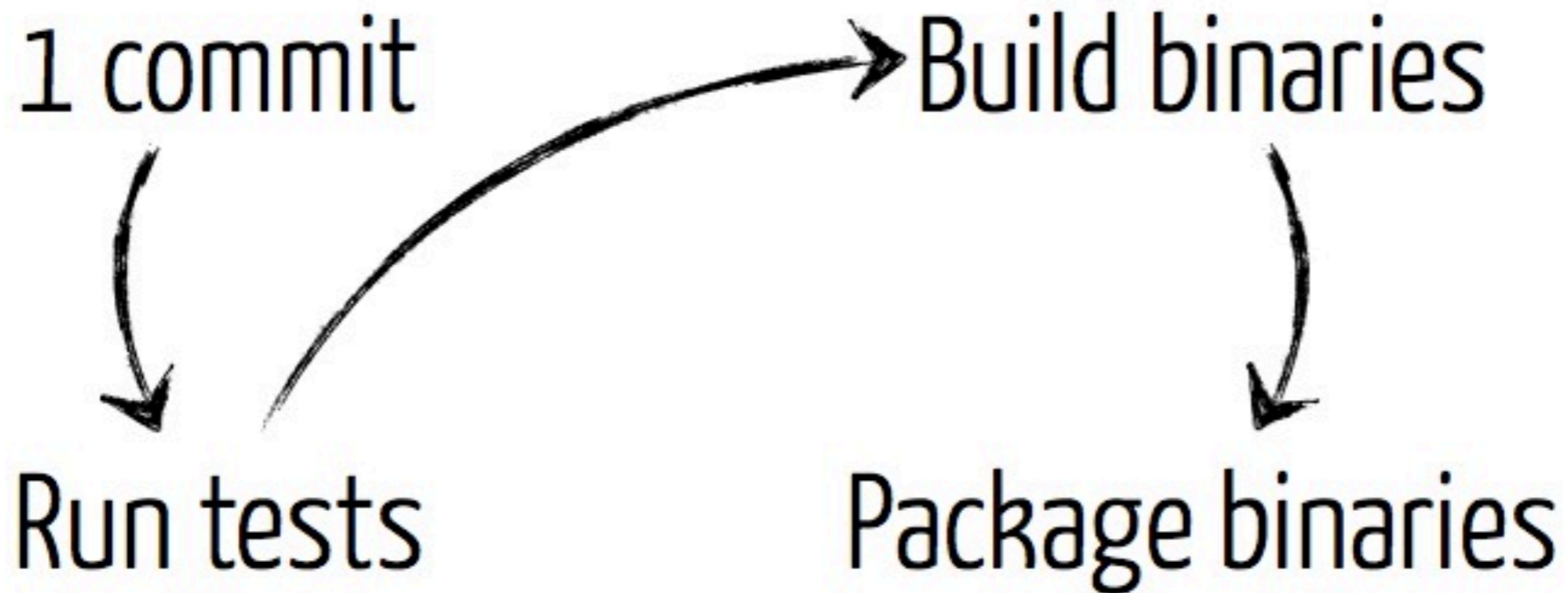


<http://www.ohloh.net/p/firefox>

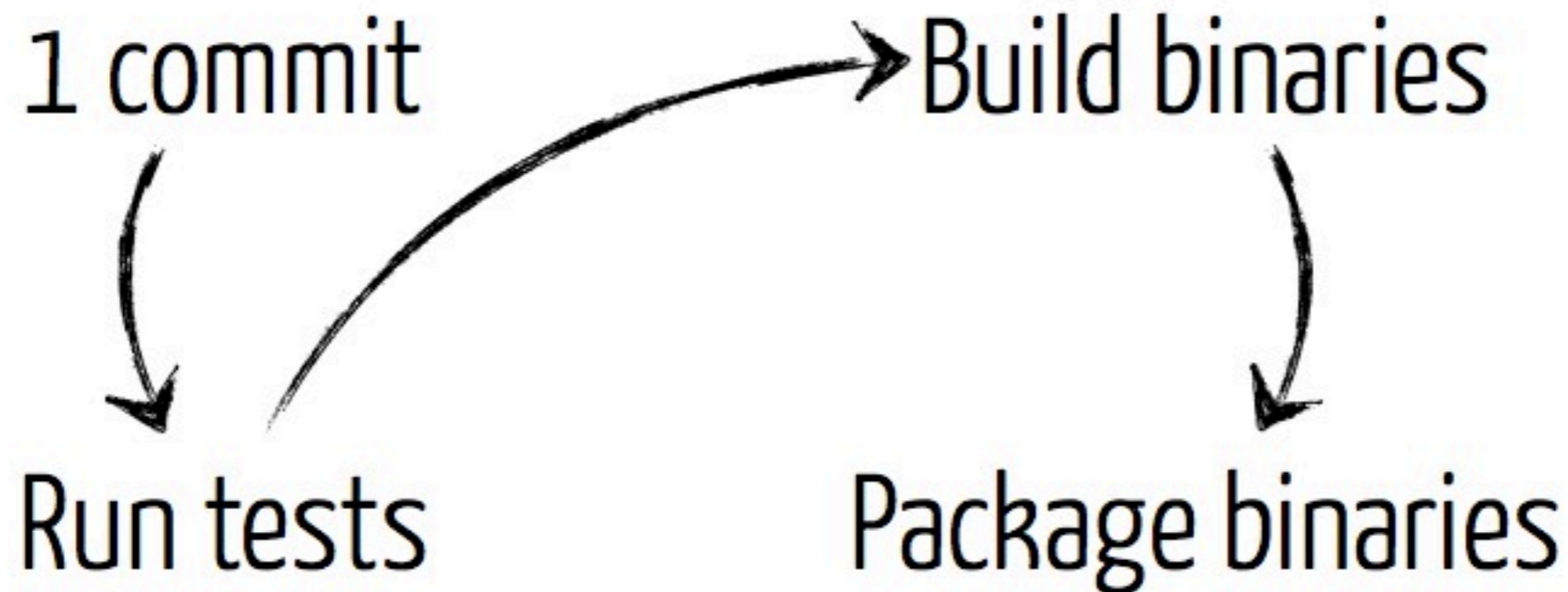
Code Comments Blanks

4

# Ok, That's Impressive... So What?



# Ok, That's Impressive... So What?



**137 hours!** (08.2012)

<http://oduinn.com/blog/2012/08/21/137-hours-compute-hours-every-6-minutes/>

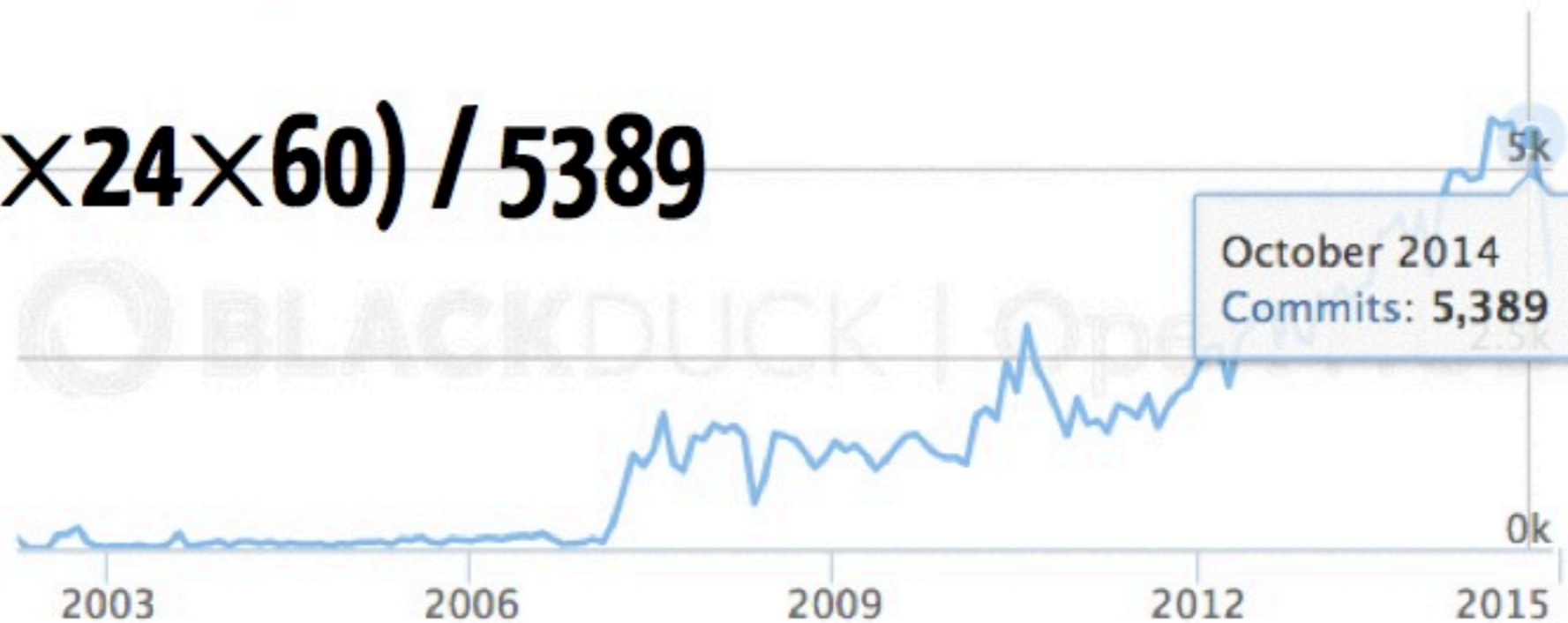


# 4403 Commits Last Month!



Commits per Month

**$(31 \times 24 \times 60) / 5389$**



**~1commit each 8 minutes**

<http://www.ohloh.net/p/firefox>

**One cannot**  
**handle it by**  
**hand!**

7

# Automatisation

- Un automate est un dispositif se comportant de manière automatique, c'est-à-dire **sans intervention d'un humain**.
- Objectif : rapidité, éviter les erreurs, se concentrer sur le code !
- « Lorsque vous écrivez une commande plus de trois fois, pensez à l'automatiser. » - Raphaël Marvie

# Quoi automatiser ?

- Création du squelette d'un projet : structure en fonction du type de projet, fichiers de configurations, etc
- Création de certains types de fichiers : « nouvelle classe » dans eclipse
- Lancement des tests
- Publication des résultats de tests
- Packaging : création d'un .jar, d'un zip, etc
- Déploiement du logiciel
- Création de la documentation
- Déploiement de la documentation

# Intégration continue ?

- **Intégration** : « Action d'incorporer un ou plusieurs éléments étrangers à un ensemble constitué, d'assembler des éléments divers afin d'en constituer un tout organique; passage d'un état diffus à un état constant; résultat de l'action. » - CNRTL
- **Continue** : « [En parlant d'une chose ayant un développement temporel] Qui dure sans interruption ou presque. » CNRTL
- **Intégration Continue** : Action d'assemblage sans interruption ou presque.

# Continuous Integration

*Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily - leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible. Many teams find that this approach leads to significantly reduced integration problems and allows a team to develop cohesive software more rapidly. This article is a quick overview of Continuous Integration summarizing the technique and its current usage.*

01 May 2006



**Martin Fowler**

**Translations:** [Portuguese](#) · [Chinese](#) · [Korean](#) · [French](#) · [Chinese](#) · [Czech](#) ·

Find **similar articles** to this by looking at these tags: [popular](#) · [agile](#) · [delivery](#) · [extreme programming](#) · [continuous integration](#)

For more information on this, and related topics, take a look at my [guide page](#) for delivery.

[ThoughtWorks](#), my employer, offers consulting and support around Continuous Integration. CruiseControl, the first

## Contents

- Building a Feature with Continuous Integration
- Practices of Continuous Integration
  - Maintain a Single Source Repository.
  - Automate the Build
  - Make Your Build Self-Testing
  - Everyone Commits To the Mainline Every Day
  - Every Commit Should Build the Mainline on an Integration Machine
  - Fix Broken Builds Immediately
  - Keep the Build Fast
  - Test in a Clone of the Production Environment
  - Make it Easy for Anyone to Get the Latest Executable
  - Everyone can see what's happening
  - Automate Deployment
- Benefits of Continuous Integration
- Introducing Continuous Integration
- Final Thoughts
- Further Reading

<http://www.martinfowler.com/articles/continuousIntegration.html>

# Continuous Integration

*Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily - leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible. Many teams find that this approach leads to significantly reduced integration problems and allows a team to develop cohesive software more rapidly. This article is a quick overview of Continuous Integration summarizing the technique and its current usage.*

---

01 May 2006



**Martin Fowler**

**Translations:** [Portuguese](#) · [Chinese](#) · [Korean](#) · [French](#) · [Chinese](#) · [Czech](#) ·

Find **similar articles** to this by looking at these tags: [popular](#) · [agile](#) · [delivery](#) · [extreme programming](#) · [continuous integration](#)

For more information on this, and related topics, take a look at my [guide page](#) for delivery.

[ThoughtWorks](#), my employer, offers consulting and support around Continuous Integration. CruiseControl, the first

## Contents

- Building a Feature with Continuous Integration
- Practices of Continuous Integration
  - Maintain a Single Source Repository.
  - Automate the Build
  - Make Your Build Self-Testing
  - Everyone Commits To the Mainline Every Day
  - Every Commit Should Build the Mainline on an Integration Machine
  - Fix Broken Builds Immediately
  - Keep the Build Fast
  - Test in a Clone of the Production Environment
  - Make it Easy for Anyone to Get the Latest Executable
  - Everyone can see what's happening
  - Automate Deployment
- Benefits of Continuous Integration
- Introducing Continuous Integration
- Final Thoughts
- Further Reading

<http://www.martinfowler.com/articles/continuousIntegration.html>

# Outils de l'intégration continue

- **Git** : système de branches, autorise le développement parallèle de features
- **Maven** : Automatisation du lancement de tâches de build, test, packaging
- **JUnit** : Lancement des tests
- **Jenkins** : Regroupement des tâches au sein d'une interface web, déclenchement de tâches sur «hook» (ex : commit)



# *maven*

- Descendant de Ant et Make
- Objectifs : Gérer les dépendances et automatiser des tâches
- Configuration d'un projet Maven : pom.xml
- Equivalent en JavaScript : Grunt pour les tâches, npm pour les dépendances

# pom.xml

```
<project>
  <!-- model version is always 4.0.0 for Maven 2.x POMs -->
  <modelVersion>4.0.0</modelVersion>

  <!-- project coordinates, i.e. a group of values which
       uniquely identify this project -->

  <groupId>com.mycompany.app</groupId>
  <artifactId>my-app</artifactId>
  <version>1.0</version>

  <!-- library dependencies -->

  <dependencies>
    <dependency>

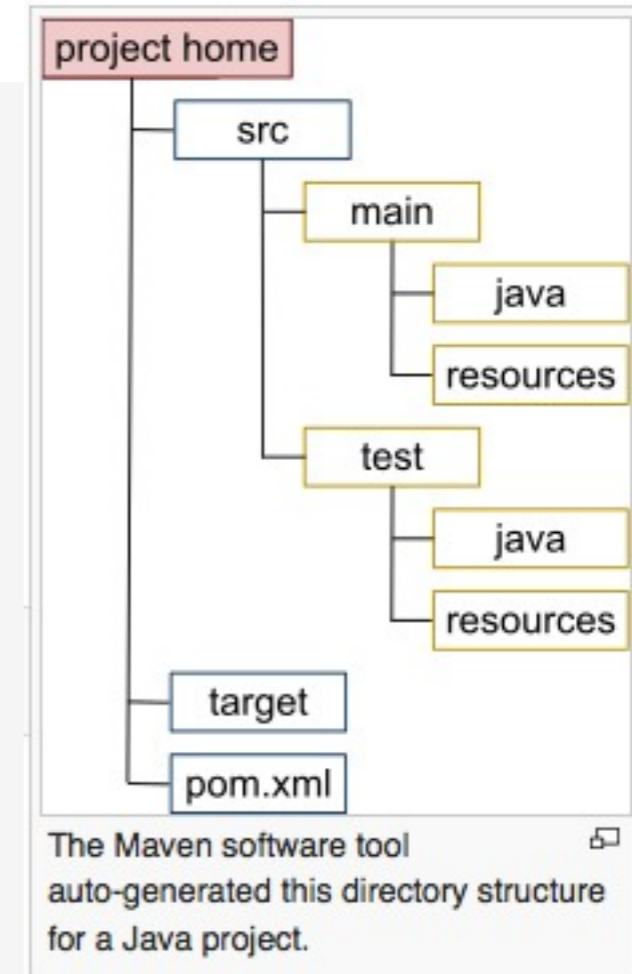
      <!-- coordinates of the required library -->

      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>

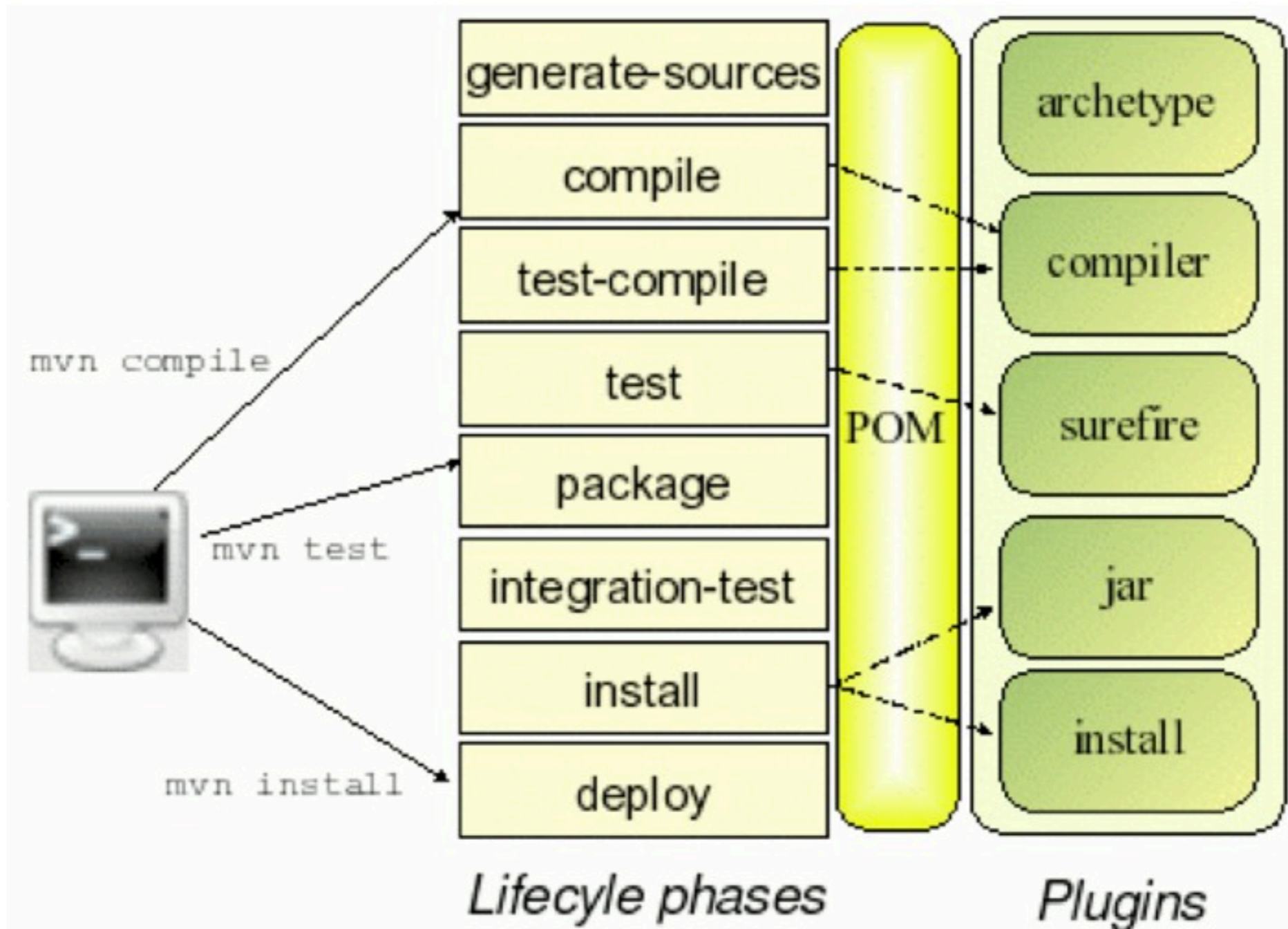
      <!-- this dependency is only used for running and compiling tests -->

      <scope>test</scope>

    </dependency>
  </dependencies>
</project>
```



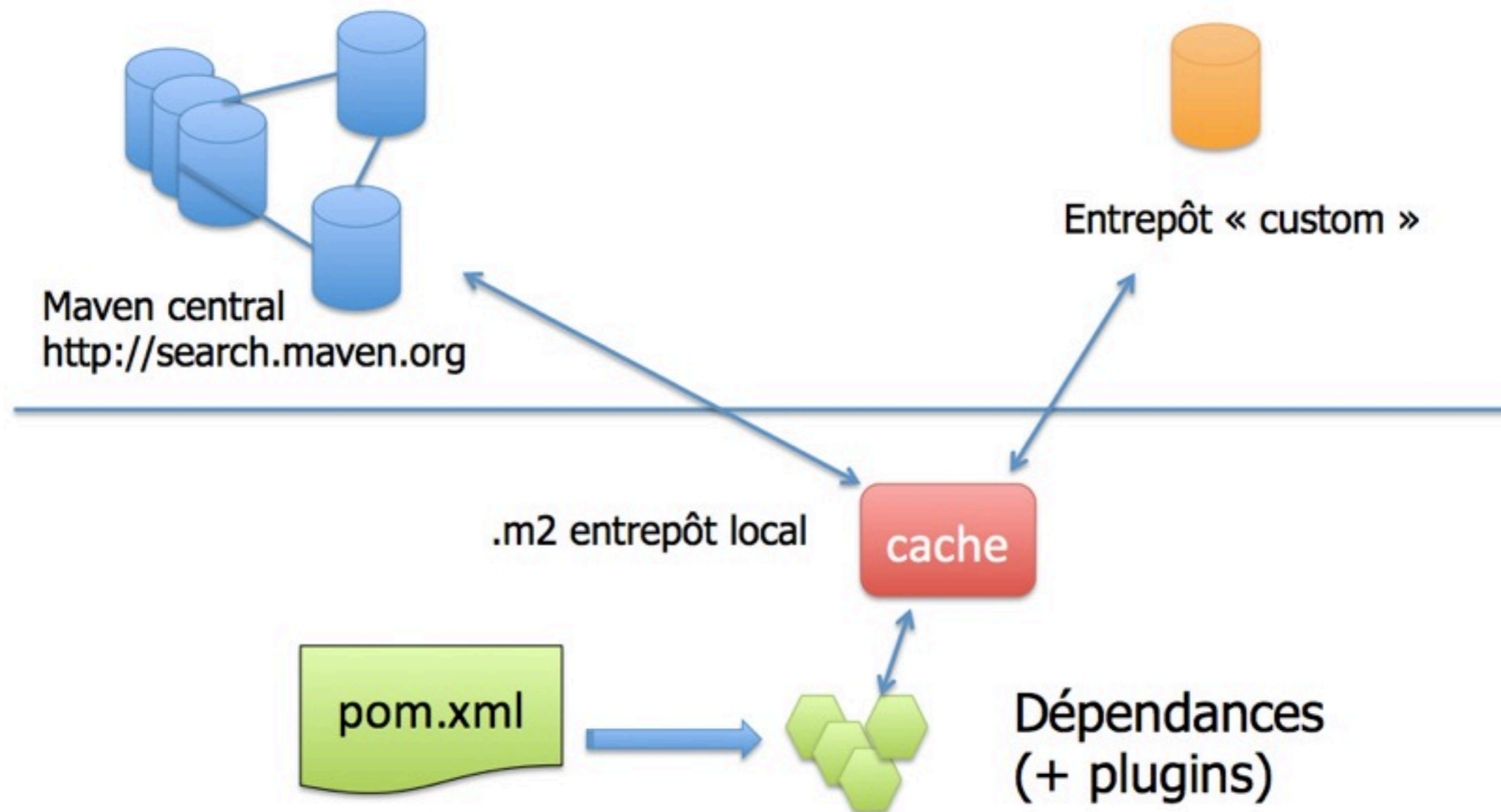
# Maven : cycle de vie par défaut pour le déploiement



# Maven goals

- mvn clean : nettoie le projet (par défaut supprime le répertoire target)
- mvn compile : lance la compilation du projet
- mvn test : lance la compilation du projet et des tests et exécute les tests
- mvn package : compile + test et créé ensuite un package (ex : .jar) contenant les binaires
- mvn install : package + installe ensuite le package créé en local
- mvn deploy : install + déploie le package créé sur un dépôt maven distant

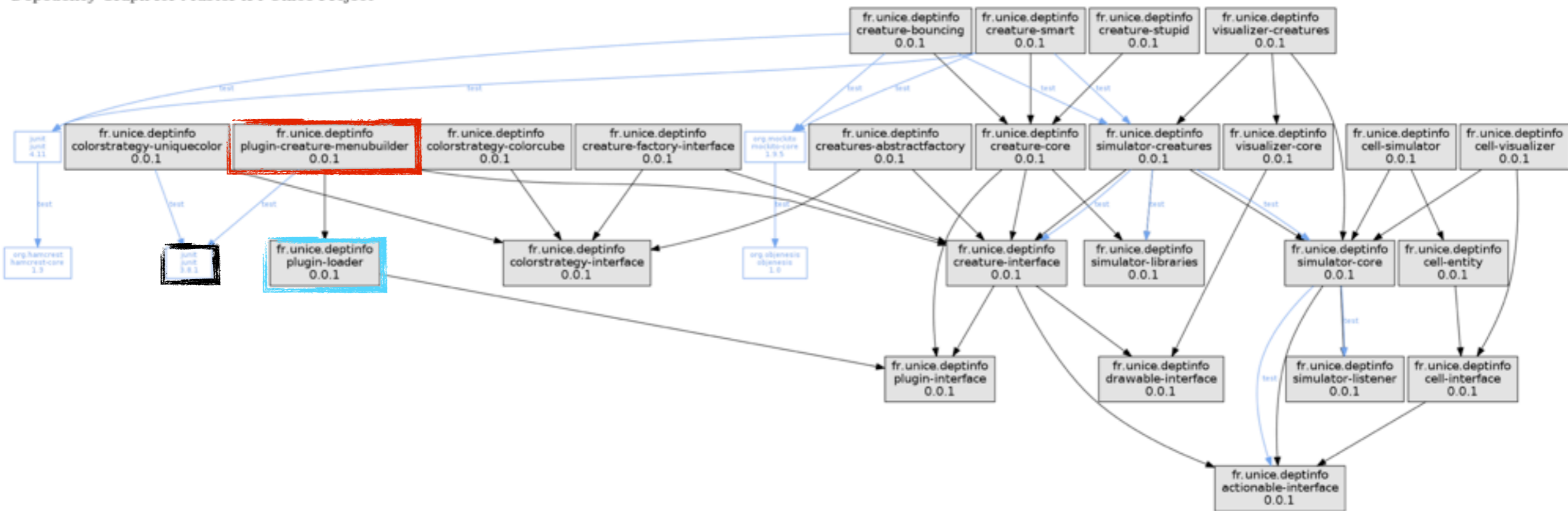
# Maven : gestion des dépôts





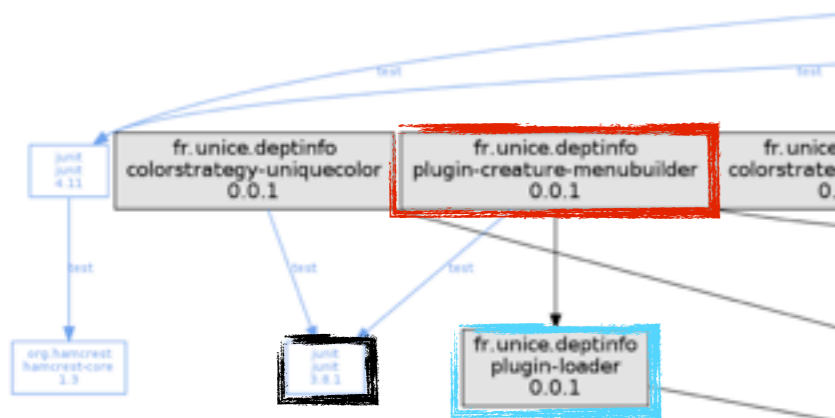
# Maven : gestion des dépendances

Dependency Graph for Master IFI Unice Project



# Maven : gestion des dépendances

Dependency Graph for Master IFI Unice Project



```
1 <?xml version="1.0"?>
2 <project xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"
3   xmlns="http://maven.apache.org/POM/4.0.0"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
5   <modelVersion>4.0.0</modelVersion>
6   <parent>
7     <groupId>fr.unice.deptinfo</groupId>
8     <artifactId>gl-ifi-parent</artifactId>
9     <version>0.0.1</version>
10  </parent>
11  <groupId>fr.unice.deptinfo</groupId>
12  <artifactId>plugin-creature-menubuilder</artifactId>
13  <version>0.0.1</version>
14  <name>plugin-creature-menubuilder</name>
15  <url>http://maven.apache.org</url>
16  <properties>
17    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
18  </properties>
19  <dependencies>
20    <dependency>
21      <groupId>junit</groupId>
22      <artifactId>junit</artifactId>
23      <version>3.8.1</version>
24      <scope>test</scope>
25    </dependency>
26    <dependency>
27      <groupId>fr.unice.deptinfo</groupId>
28      <artifactId>creature-interface</artifactId>
29      <version>0.0.1</version>
30    </dependency>
31    <dependency>
32      <groupId>fr.unice.deptinfo</groupId>
33      <artifactId>plugin-loader</artifactId>
34      <version>0.0.1</version>
35    </dependency>
36  </dependencies>
37 </project>
```





# Jenkins

Tous												
all-continuous												
all-scratch												
dashboard												
service-continuous												
+												
S	M	Nom du projet ↓	Dernier succès	Dernier échec	Dernière durée		Number of builds					
		<a href="#">admin-develop-continuous</a>	10 j - <a href="#">#8</a>	2 mo. 0 j - <a href="#">#2</a>	2 mn 43 s			6		0		2
		<a href="#">admin-develop-scratch</a>	29 j - <a href="#">#1</a>	s. o.	1 h 7 mn			1		0		0
		<a href="#">backend-develop-continuous</a>	2 mo. 0 j - <a href="#">#13</a>	9 j 0 h - <a href="#">#18</a>	1 mn 33 s			5		0		5
		<a href="#">backend-develop-scratch</a>	29 j - <a href="#">#3</a>	s. o.	3 mn 33 s			3		0		0
		<a href="#">client-develop-continuous</a>	1 mo. 9 j - <a href="#">#47</a>	8 j 8 h - <a href="#">#51</a>	1 mn 14 s			6		0		4
		<a href="#">client-develop-scratch</a>	29 j - <a href="#">#3</a>	s. o.	5 mn 53 s			3		0		0
		<a href="#">core-backend-develop-continuous</a>	8 j 7 h - <a href="#">#15</a>	2 mo. 17 j - <a href="#">#8</a>	39 s			8		0		2
		<a href="#">core-backend-develop-scratch</a>	29 j - <a href="#">#3</a>	s. o.	4 mn 5 s			3		0		0
		<a href="#">core-develop-continuous</a>	2 mo. 1 j - <a href="#">#7</a>	2 mo. 17 j - <a href="#">#4</a>	26 s			4		0		3
		<a href="#">core-develop-scratch</a>	29 j - <a href="#">#3</a>	s. o.	1 mn 33 s			3		0		0
		<a href="#">service-calendar-develop-continuous</a>	8 j 11 h - <a href="#">#10</a>	2 mo. 17 j - <a href="#">#1</a>	49 s			9		0		1
		<a href="#">service-calendar-develop-scratch</a>	29 j - <a href="#">#3</a>	s. o.	2 mn 49 s			3		0		0
		<a href="#">service-facebook-develop-continuous</a>	8 j 11 h - <a href="#">#8</a>	s. o.	52 s			8		0		0



# Jenkins

- [Retour au tableau de bord](#)
- [État](#)
- [Modifications](#)
- [Répertoire de travail](#)
- [Lancer un build](#)
- [Supprimer Projet](#)
- [Configurer](#)
- [Coverage Report](#)
- [Log du dernier accès à Git](#)

## Projet backend-develop-continuous

Continuous build of backend module.  
Triggered by hooks.

[modifier la description](#)

Désactiver le projet

[Coverage Report](#)

[Espace de travail](#)

[Changements récents](#)

[Derniers résultats des tests \(3 échecs / ±0\)](#)

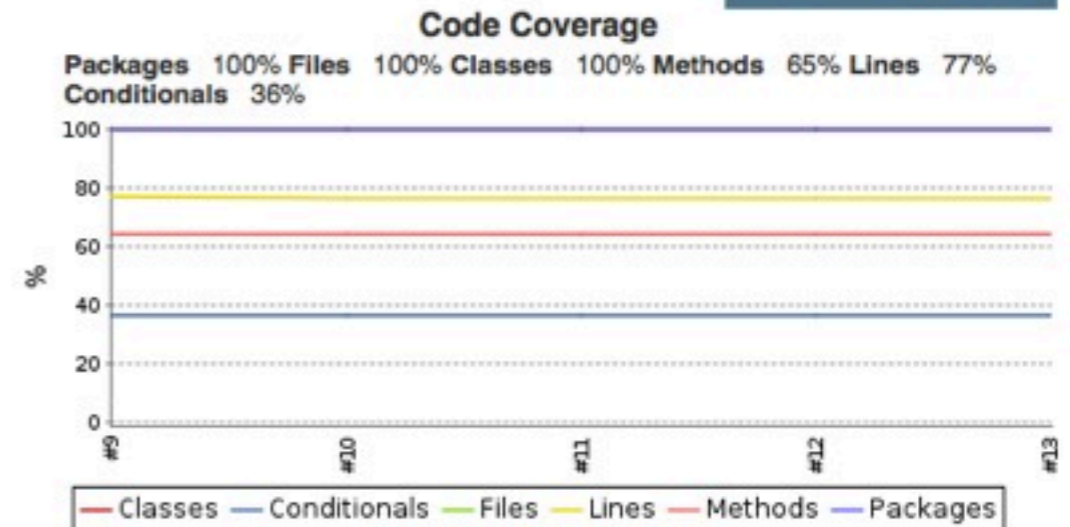
### Liens permanents

- [Dernier build \(#18\), il y a 9 j 0 h](#)
- [Dernier build stable \(#13\), il y a 2 mo. 0 j](#)
- [Dernier build avec succès \(#13\), il y a 2 mo. 0 j](#)
- [Dernier build en échec \(#18\), il y a 9 j 0 h](#)
- [Dernier build non réussi \(#18\), il y a 9 j 0 h](#)

Historique des builds		tendance
#18	10 oct. 2015 21:46	
#17	9 oct. 2015 15:49	
#16	27 sept. 2015 14:09	
#15	24 sept. 2015 16:29	
#14	24 sept. 2015 11:43	
#13	20 août 2015 19:32	
#12	19 août 2015 21:46	
#11	12 août 2015 18:06	
#10	12 août 2015 12:02	
#9	11 août 2015 15:20	

[RSS des builds](#) [RSS des échecs](#)

[i.the6thscreen.fr/job/backend-develop-continuous/cobertura/](http://i.the6thscreen.fr/job/backend-develop-continuous/cobertura/)





# Jenkins

## Gestion de code source

- Aucune
  - CVS
  - CVS Projectset
  - Git
- Repositories

Repository URL  [?](#)

Credentials

## Branches to build

Branch Specifier (blank for 'any')  [?](#)



# Jenkins

Scrutation de l'outil de gestion de version [?](#)

Planning

@daily

Aurait été lancé à lundi 19 octobre 2015 00 h 00 CEST; prochaine exécution à mardi 20 octobre 2015 00 h 00 CEST.

Ignore post-commit hooks  [?](#)

## Environnements de Build

Ajouter le répertoire bin/ de Node/npm au PATH

Installation

Default

Specify needed nodejs installation where npm installed packages will be provided to the PATH

Configure release build

Generate Release Notes

## Build

Exécuter un script shell [?](#)

Commande `cd ./backend && chmod +x ./jenkins.sh && ./jenkins.sh ../core-backend ../core`



# Jenkins

## Publier le rapport des résultats des tests JUnit



XML des rapports de test

Une configuration du type [Fileset 'includes'](#) qui indique où se trouvent les fichiers XML des rapports de test, par exemple 'myproject/target/test-reports/\*.xml'. Le répertoire de base (basedir) du fileset est [la racine du workspace](#).

Retain long standard output/error



Health report amplification factor



1% failing tests scores as 99% health. 5% failing tests scores as 95% health

Supprimer

## Publish Cobertura Coverage Report

Cobertura xml report pattern

This is a file name pattern that can be used to locate the cobertura xml report files (for example with Maven2 use **\*\*/target/site/cobertura/coverage.xml**). The path is relative to the module root unless you have configured your SCM with multiple modules, in which case it is relative to the workspace root. Note that the module root is SCM-specific, and may not be the same as the workspace root. Cobertura must be configured to generate XML reports for this plugin to function.

Avancé...

Supprimer

## Notifier par email



Destinataires

Liste des destinataires, séparés par un espace. Un email sera envoyé lors d'un échec d'un build.

Envoyer un email pour chaque build instable



# Jenkins

Wipe out repository & force clone



Supprimer

Ajouter

Delete SCM

Add SCM

Subversion

## Ce qui déclenche le build

- Déclencher les builds à distance (Par exemple, à partir de scripts)
- Build when a change is pushed to BitBucket
- Build when we receive a notification from Rundeck
- Construire après le build sur d'autres projets



Projet à surveiller

core-develop-scratch,

- Déclencher que si la construction est stable
- Déclencher même si la construction est instable
- Déclencher même si la construction échoue