# INTÉGRATION CONTINUE

Intégration continue S. Fauvel – Univ. Nantes 2023

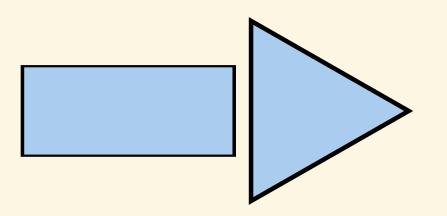
#### **PLAN**

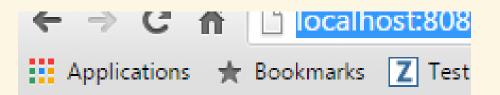
- Intégration
- Intégration continue
- Jenkins

## INTÉGRATION

```
public StoreServer(PriceSto
    this.priceStorage = pri
    Basket basket = new Bas
    this.shopping = new Sho
}

public void start(int port)
    server = HttpServer.cre
    server.createContext("/
```





#### **Grocery Store**

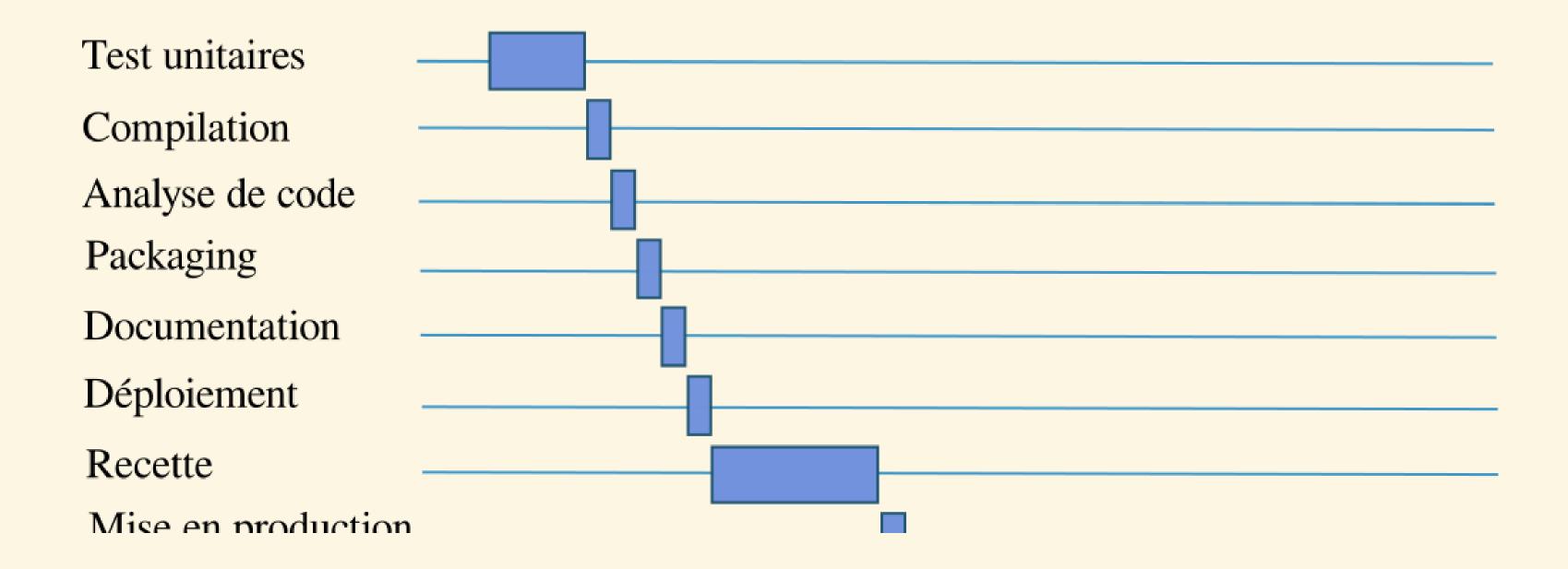
Cliquer sur le fruit que vous voulez a

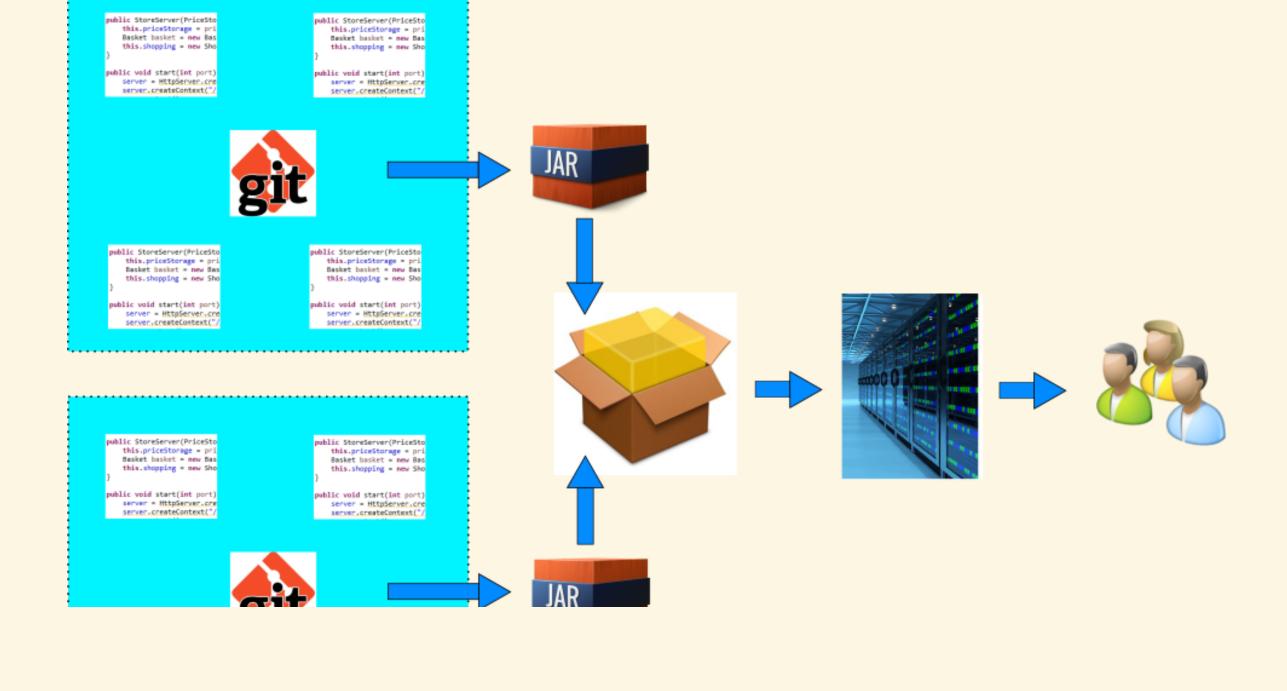
- apple (2 euros)
- banana (3 euros)

Montant du panier: 10 euros

Contenu du panier:

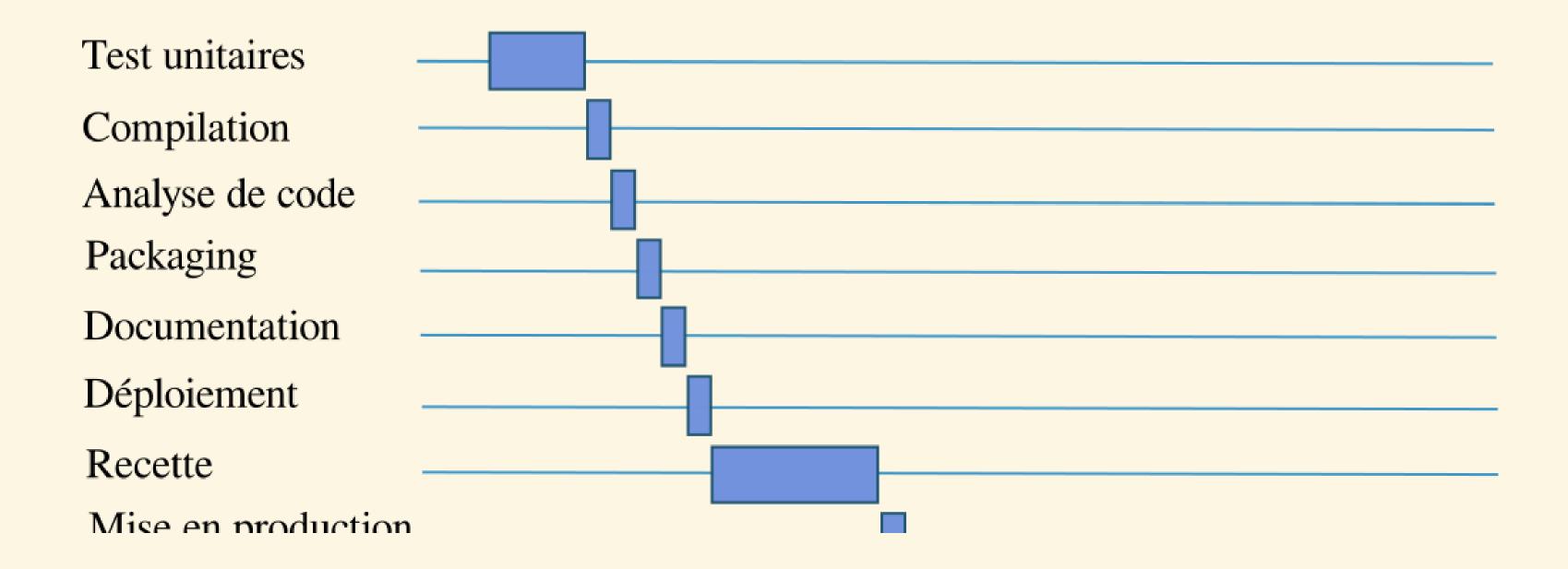
apple

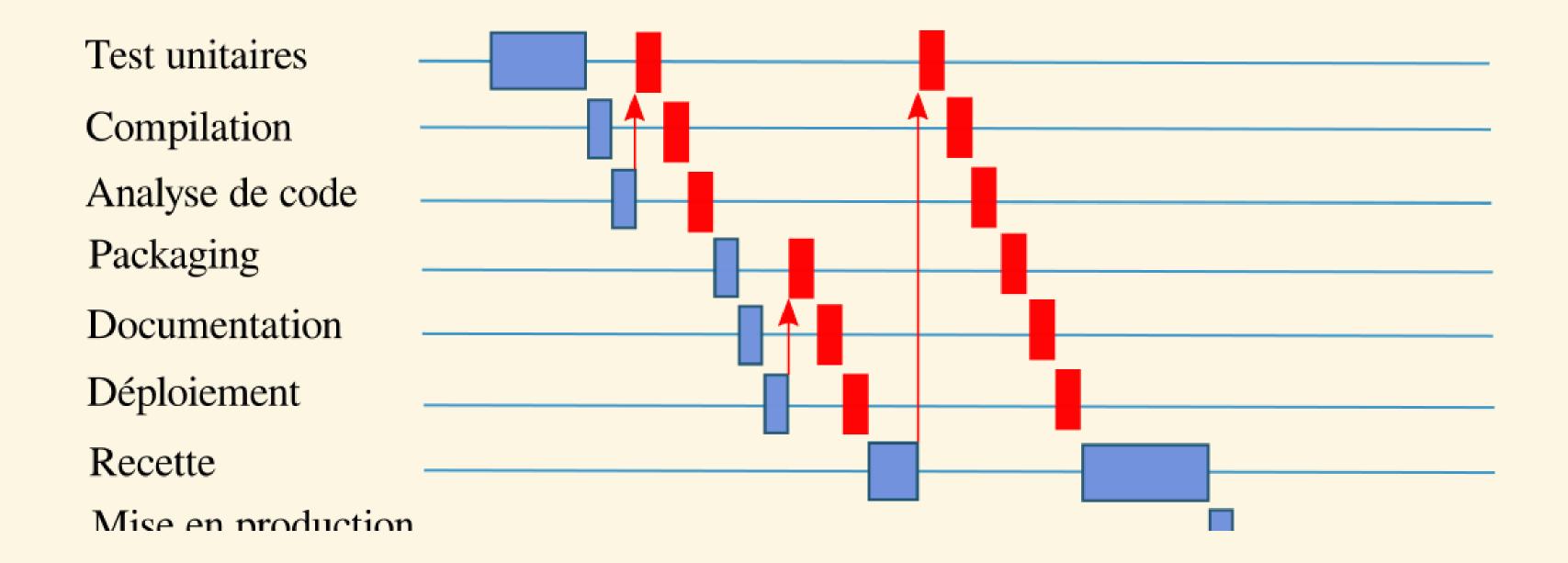


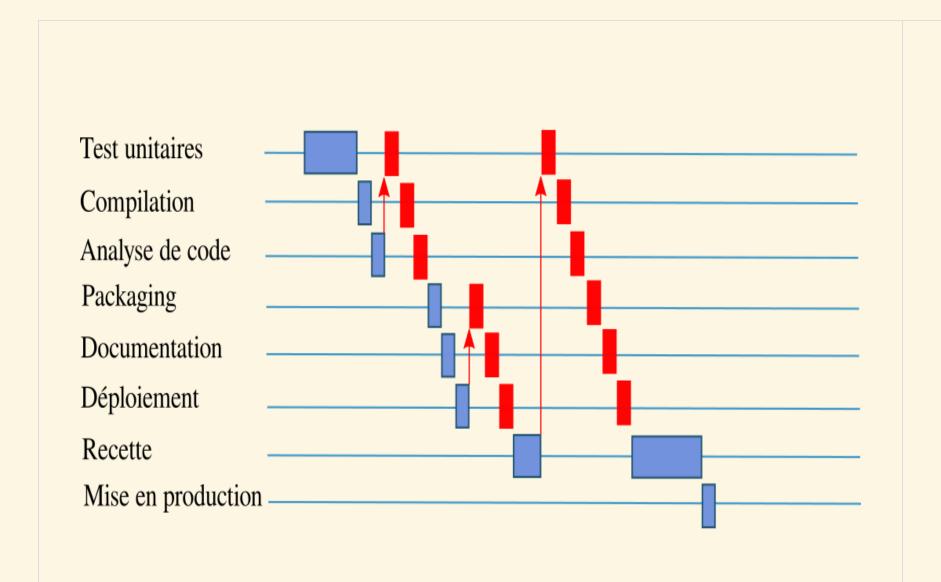


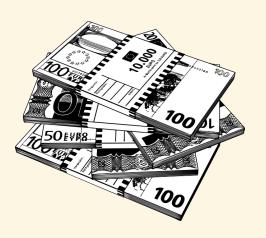
#### **PLANNING**

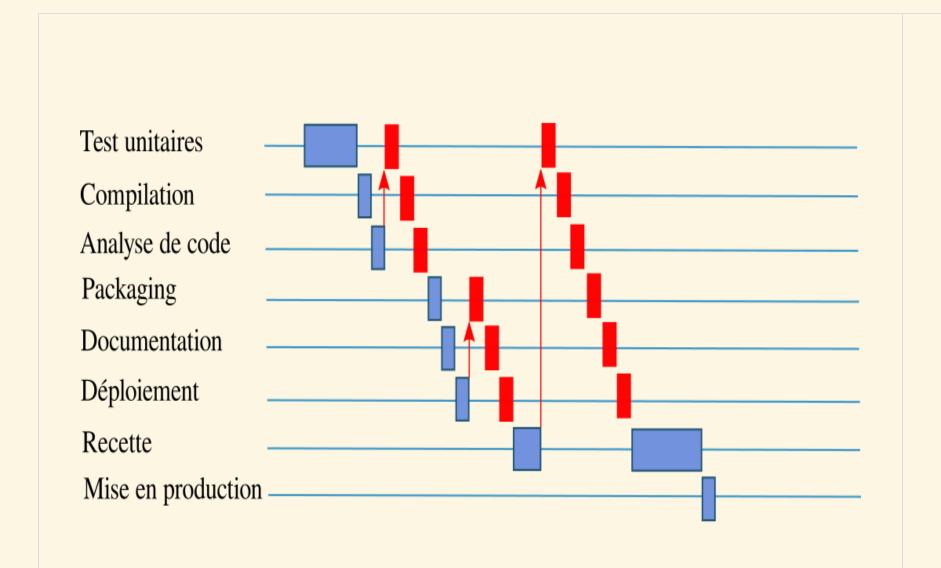




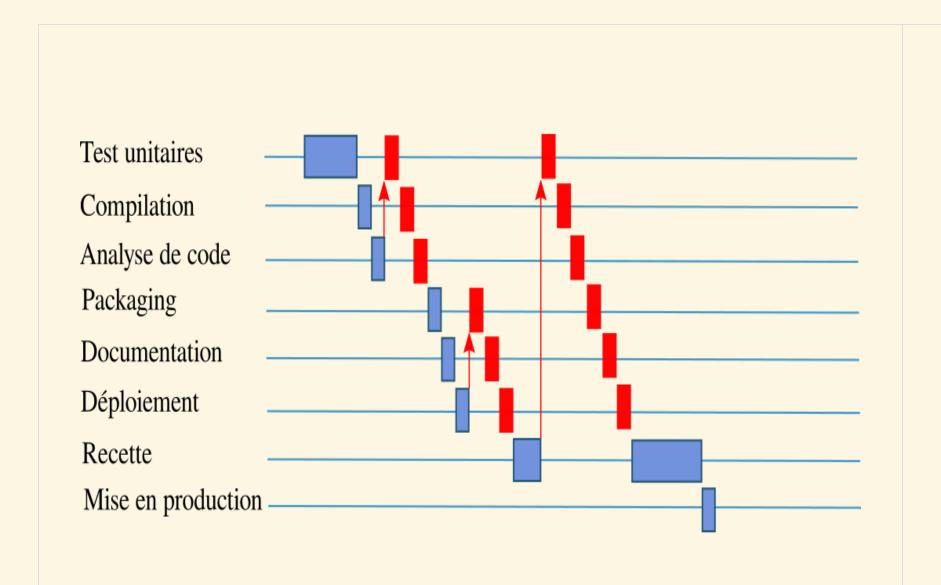


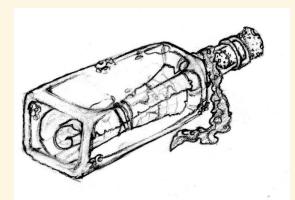


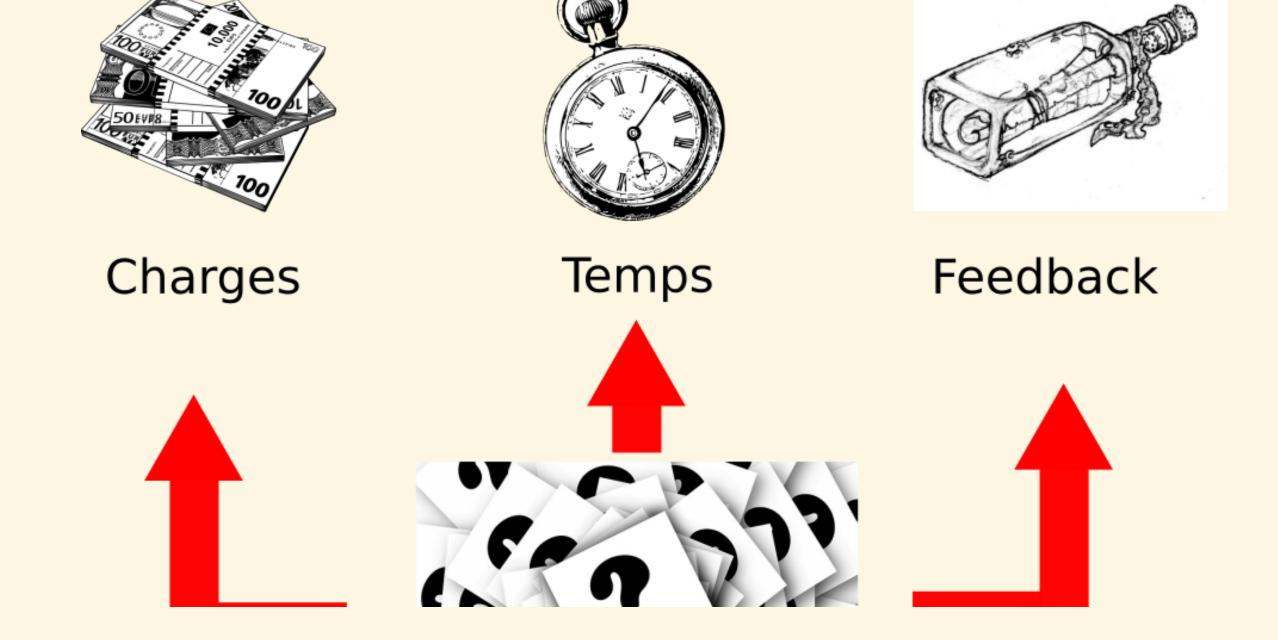












#### 1 CONSTAT

- L'intégration est une phase compliquée
- C'est une étape incertaine
- Il faut prévoir une charge de travail

#### INTÉGRATION DANS UN CYCLE EN V

- Le moins souvent possible
- 1 seul grande intégration
- Estimer et planifier la phase d'intégration
- Prévoir pour limiter les risques

#### INTÉGRATION DANS UNE APPROCHE AGILE

- Le plus souvent possible
- De nombreuses petites intégrations
- Incorporer cette phase au développement
- Avoir une feedback au plus tôt

## INTÉGRATION CONTINUE

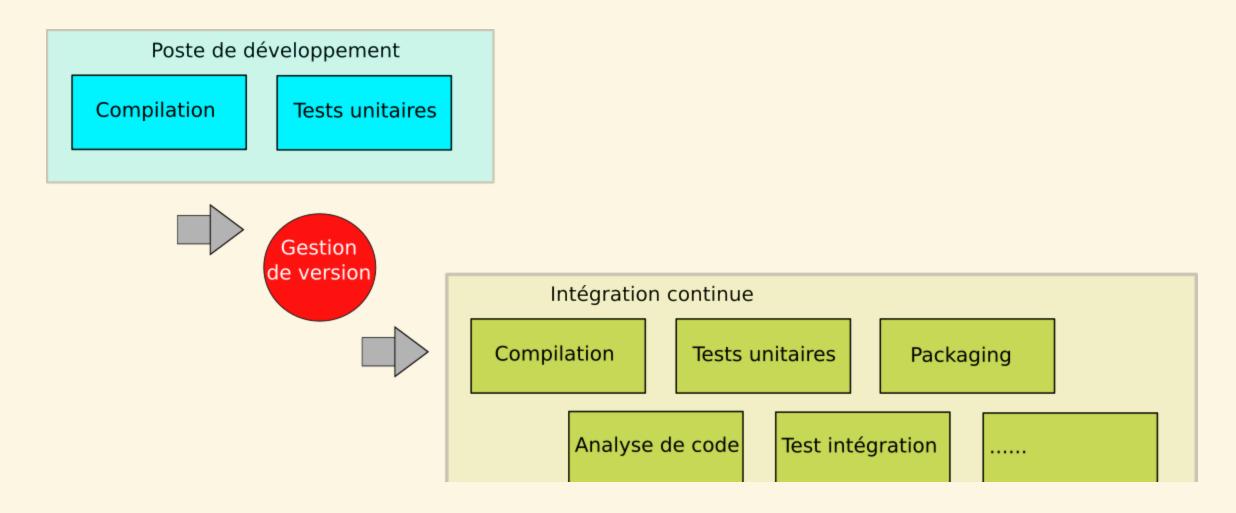
#### UKIGINE

- Grady Booch Object Oriented Design: With Applications, 1991
- Kent Beck Extreme Programming Explained, 1999
  - Diminuer les risques de conflit
  - Détecter les défauts au plus tôt
  - Avoir un logiciel opérationnel en permanence
- Martin Fowler Continuous Integration, 2000

http://www.martinfowler.com/articles/originalContinuousIntegration.html

#### INTILUNATION CONTINUL

Automatiser la chaîne de build pour faire l'intégration de manière transparente



#### **BONNES PRATIQUES**

- Commiter régulièrement du code fonctionnel
  - L'intégration continue s'appuie sur le code commité
  - L'application doit compiler et les tests passer
- Avoir des tests automatiser pour détecter les anomalies
- Corriger les builds en échec
- Optimiser les temps de builds pour améliorer le feedback

- Réduction des risques lié à l'incertitude
  - Etat du projet connu en permanence
  - Dernière version stable identifiée
- Réduction des tâches répétitives
  - Gain de temps pour les équipes
  - Réduction des délais pour l'intégration
- Impose des bonnes pratiques
  - Oblige une rigueur de travail

#### DÉFINITIONS

- Continuous Integration: Vérifie l'intégration
- Continuous Delivery: Prêt à être mis en production
- Continuous Deployment: Mise en production

https://medium.com/jorgeacetozi/continuous-integration-vs-continuous-delivery-vs-continuous-deployment-d5839a85a959

#### OUTILS

- Jenkins
- TeamCity
- Travis
- Gitlab Cl
- GitHub Actions

• • • •

### **JENKINS**



#### Serveur d'intégration continue open source Gère et contrôle les tâches du cycle de vie d'un logiciel:

- Compilation
- Documentation
- Tests
- Packaging
- analyse de code
- • •

#### JENKINS - INSTALLATION

Serveur Web, un simple war

Lancement:

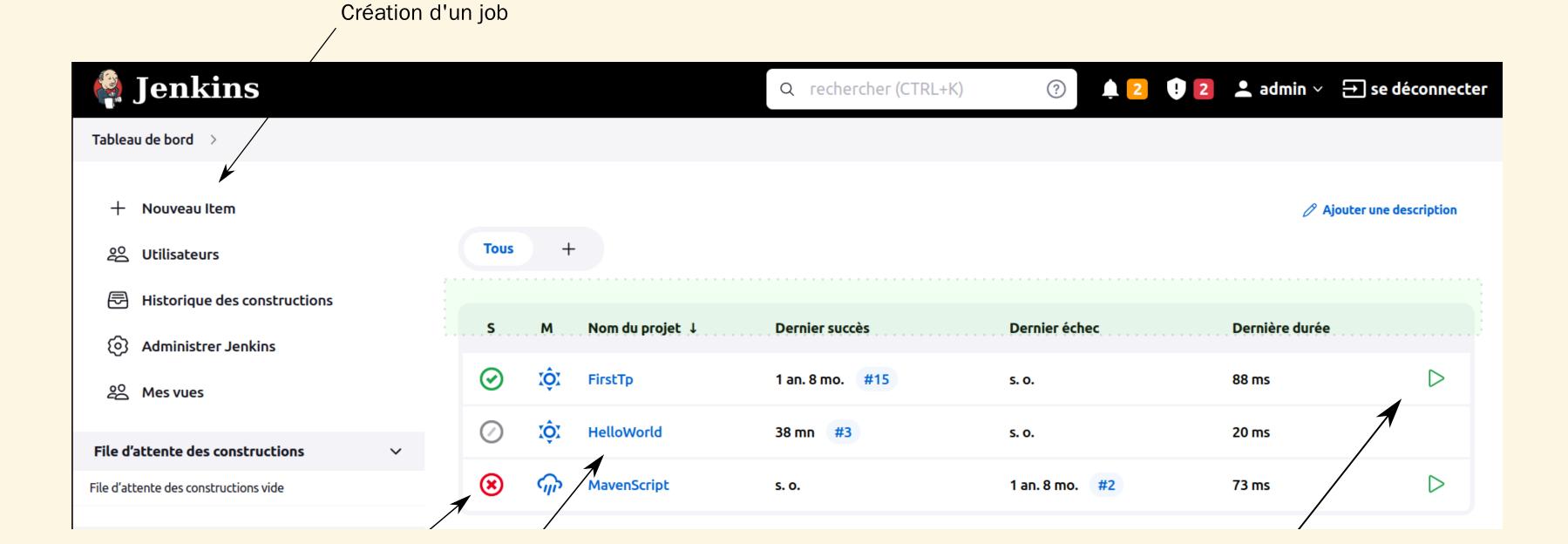
java -jar jenkins.war

Accès:

http://localhost:8080

#### JENKINS - JOBS

- Un job est une configuration pour l'intégration d'un projet
- Un projet peut avoir plusieurs jobs
- Le lancement d'un job exécute les instructions décrites dans sa configuration



### JENKINS - STATUT DU JOB

<b>⊘</b>	Succès	Tout fonctionne
<b>※</b>	En échec	La construction n'a pas pu se faire (compilation, …)
(!)	Instable	La build a abouti mais des tests échouent
0	Désactivé	Le job est désactivé

#### JENKINS - JOBS

#### Plusieurs types de jobs

- Free-style : accès à toutes les options disponibles
- Maven : basé sur une configuration Maven
- Multi-configuration: projets complexes avec plusieurs builds
- Job externe : monitoring de processus externe
- Pipeline : définition du processus d'intégration/déploiement sous forme de script



34							
		ÐΙ	n	₹	П	ĸ	9
VIS.	J	<u> </u>	7.7	••	7	-	

Configuration

General

General

Description

Ce qui déclenche le build

Environnements de Build

Environnements de Build

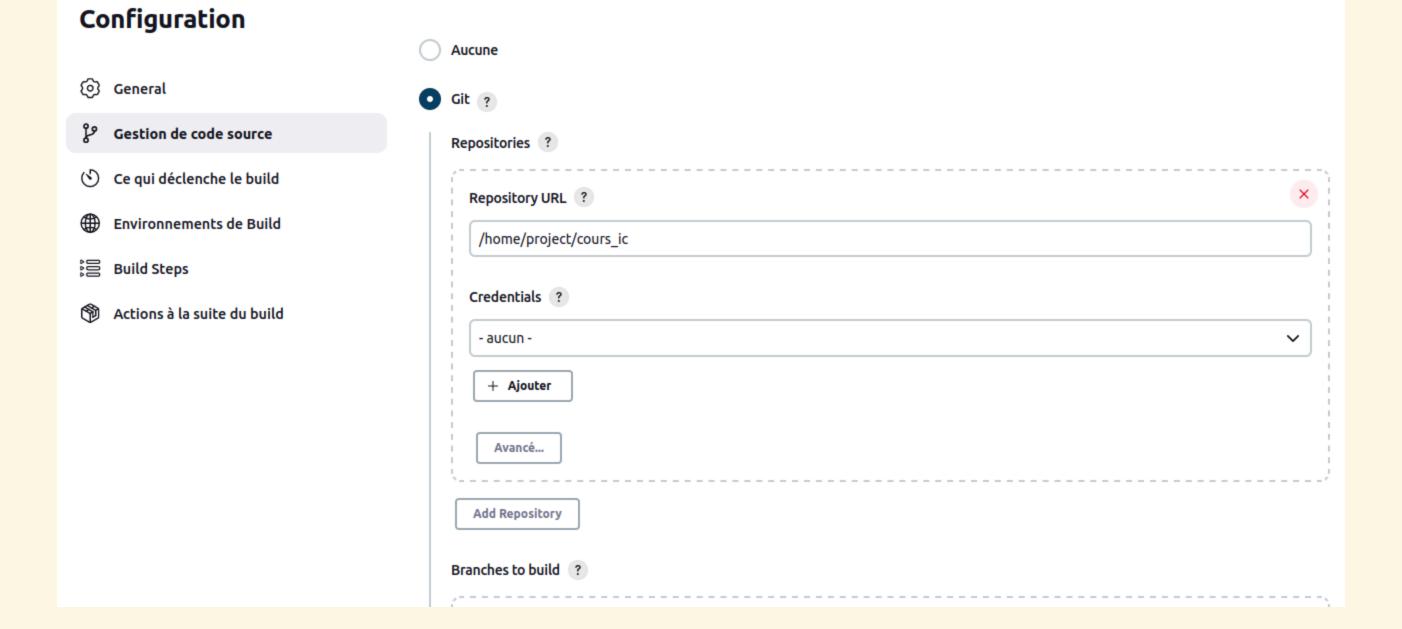
Build Steps

Actions à la suite du build

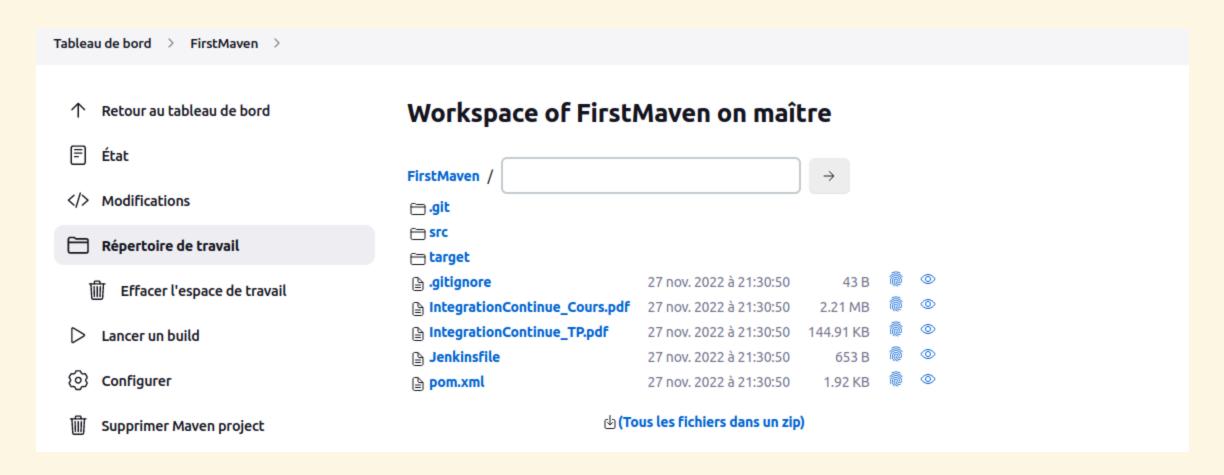
Exécuter des bui

Exécuter des bui

Configuration	General	Enabled 🕢
<b>⊚</b> General	Description	
မှ Gestion de code source		
Ce qui déclenche le build		
Environnements de Build	[Plain text] Prévisualisation	/is
₿ Build Steps	Ce build a des paramètres ?	
Actions à la suite du build	Supprimer les anciens builds ?	
	Throttle builds ?	
	Exécuter des builds simultanément si nécessaire ?	
	Avancé	



- Récupération dans un espace dédié
- Construction de l'application dans cet espace



Configuration	Ce qui déclenche le build
General  Gestion de code source  Ce qui déclenche le build  Environnements de Build  Pre Steps  Build  Post Steps	Lance un build à chaque fois qu'une dépendance SNAPSHOT est construite ?  Déclencher les builds à distance (Par exemple, à partir de scripts) ?  Construire après le build sur d'autres projets ?  Construire périodiquement ?  Scrutation de l'outil de gestion de version ?  Planning ?  H/15 * * * *
O Configuration du build	Aurait été lancé à Sunday, November 27, 2022 at 9:30:25 PM Coordinated Universal Time; prochaine exécution à Sunday, November 27, 2022 at 9:45:25 PM Coordinated Universal Time.
Actions à la suite du build	Ignore post-commit hooks ?
	Envisonaments de Puild

- Planification:
  - Construire périodiquement
  - Scruter la gestion de configuration
- La syntaxe est celle du Cron
  - [MINUTES] [HEURES] [JOURMOIS] [MOIS] [JOURSEMAINE]
  - '\*' représente l'ensemble des valeurs possibles
  - 'N' déclenche lorsque la valeur vaut N
  - '\*/X' déclenche tous les X
- Exemple:

<b>⊘</b> <u>#12</u>	22 mars 2021 11:07
<b>⊘</b> <u>#11</u>	22 mars 2021 11:06
<b>⊗</b> #10	22 mars 2021 10:59
<b>⊘</b> <u>#9</u>	22 mars 2021 10:56

## Dernier accès au gestionnaire de configuration



### Dernier Log du dernier accès à Git

Started on Nov 27, 2022, 9:42:00 PM

Using strategy: Default

[poll] Last Built Revision: Revision b2c696c6df97f54c1e2ce80e9ede62d4d4c36493 (refs/remotes/origin/master)

The recommended git tool is: NONE

## ENVIRONNEMENT DE BUILD

#### Environnements de Build



# BUILD

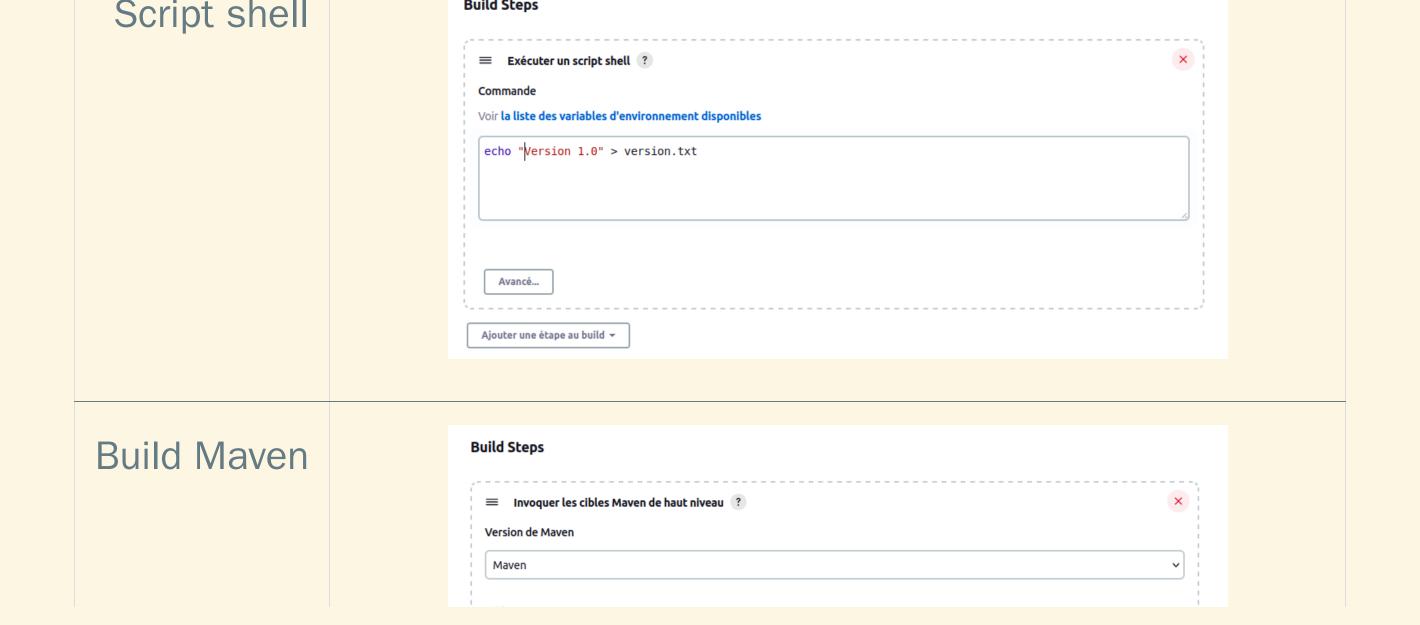
### **Build Steps**

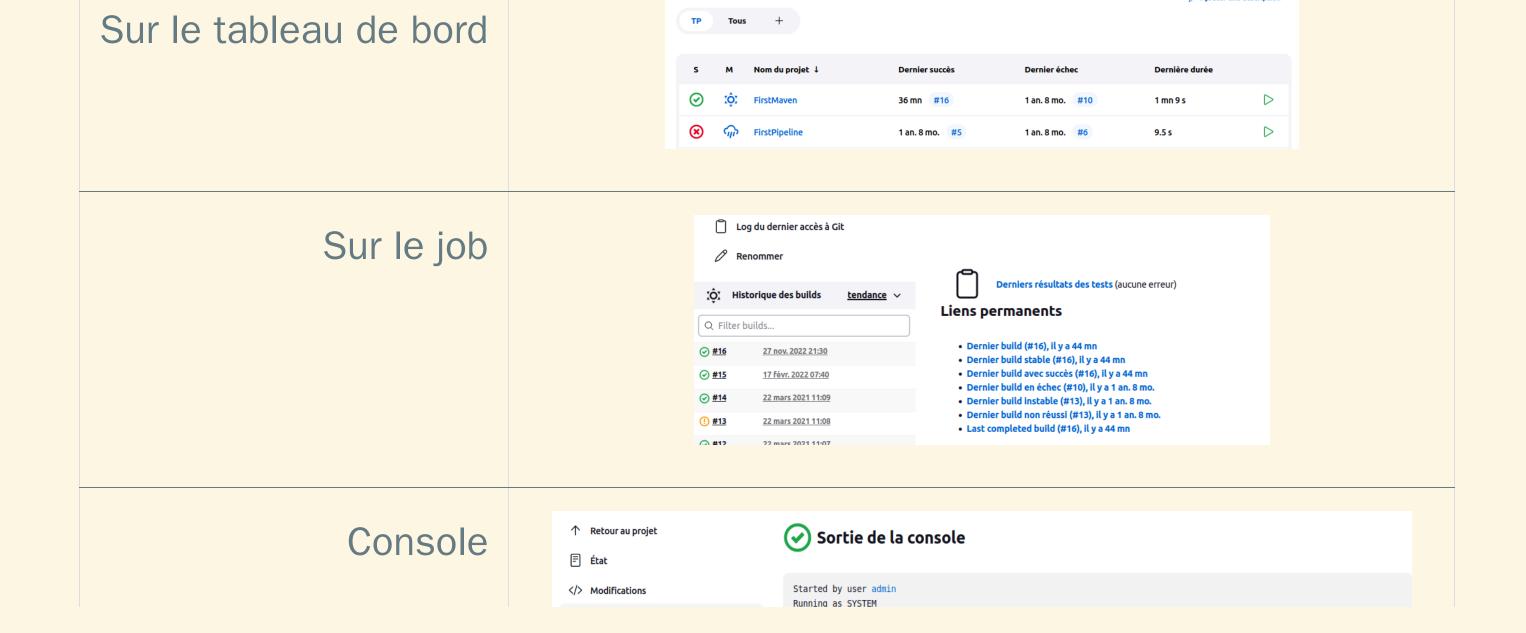
Ajouter une étape au build 🔺

Exécuter un script shell

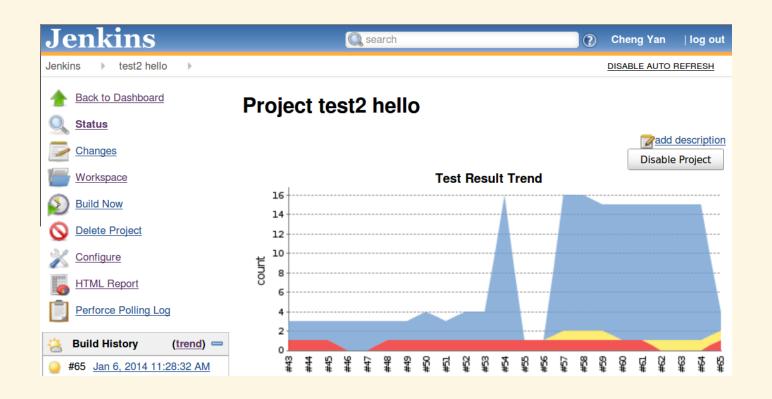
Exécuter une ligne de commande batch Windows

Invoquer les cibles Maven de haut niveau

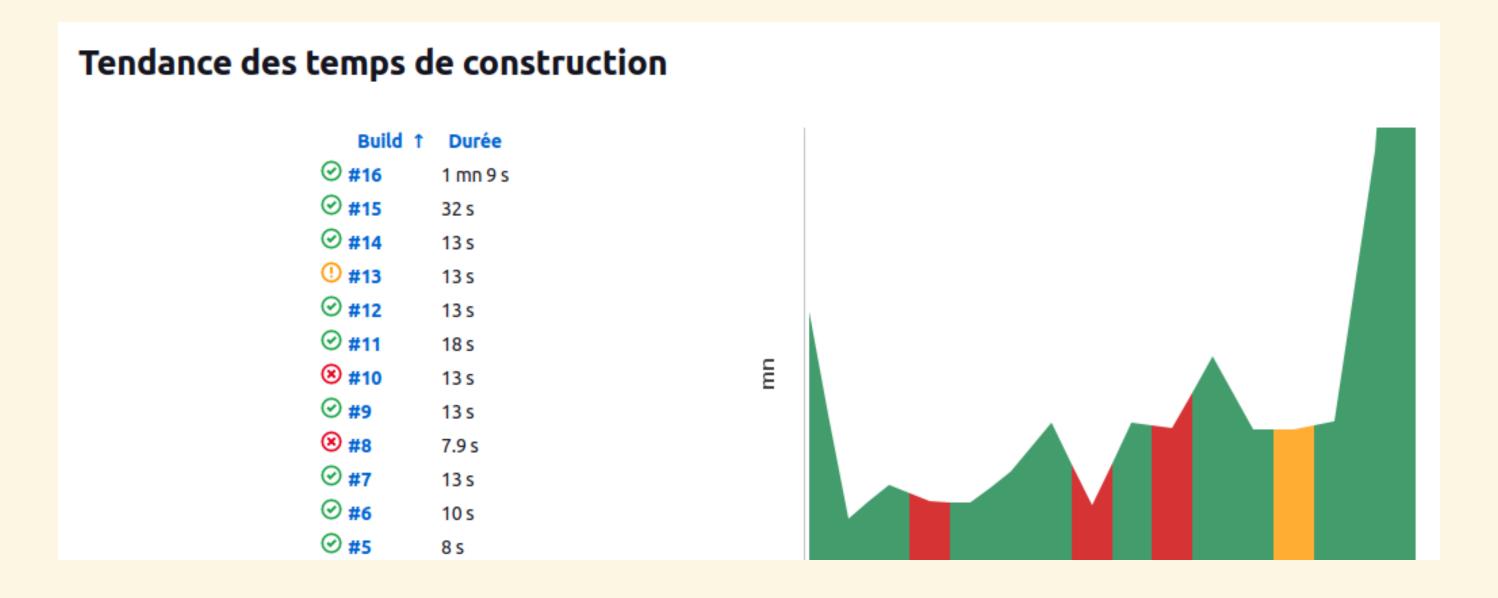




- En cas d'échec des tests, l'état du build devient instable
- Le résultat des tests est affiché
- On peut accéder au rapport détaillé d'exécution des tests



Visualisation du temps de construction du build



# ACTIONS À LA SUITE DU BUILD

#### Filter

Consolider les résultats des tests en aval

Archiver des artefacts

Construire d'autres projets (projets en aval)

Déployer les artefacts dans le repository Maven

Publish HTML reports

Enregistrer les empreintes numériques des fichiers pour en suivre l'utilisation

Git Publisher

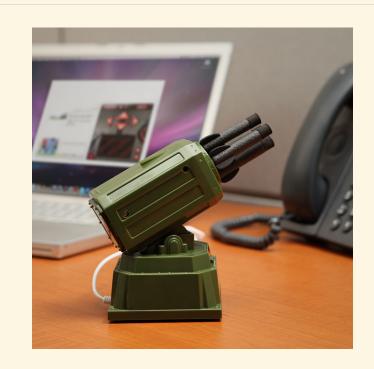
Ajouter une action après le build -

Il est possible (recommandé) d'envoyer une notification en cas d'échec du build L'échec d'un build indique un problème qu'il faut régler au plus vite.

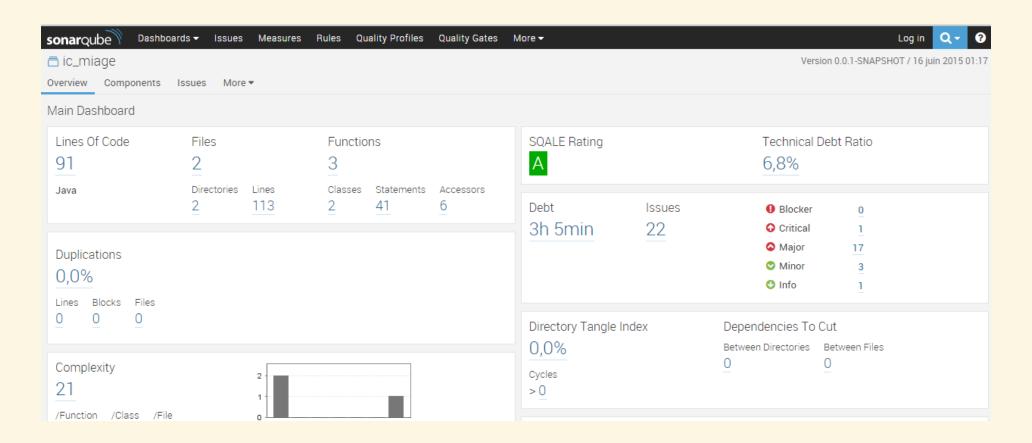
La notification ne doit pas être considérée comme une simple information mais comme une alerte.

D'autres moyens peuvent être utilisés pour signaler le problème





- Dautres actions peuvent etre realisées en plus de la construction
  - On peut déclencher une analyse de code (Sonar, Checkstyle, ···)
  - Le rapport d'analyse est généré et accessible via des plugins



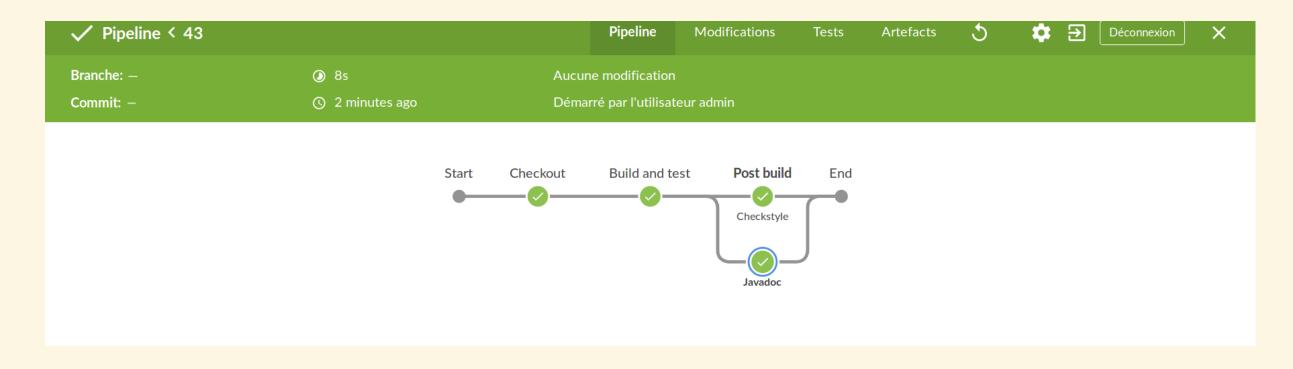
### JENNINO – OKRANIOALIUN DEO DUILDO

- La multiplication des analyses augmente le temps de traitement et la charge du serveur d'intégration
- Certaines analyses ont moins d'intérêt à être réalisées en continue
- Créer plusieurs jobs avec des objectifs et des fréquences différents
- Exemple:
  - En continue : compilation, tests unitaires
  - Quotidien : analyse de code, javadoc, test bout en bout
  - Hahdamadaira: tact de charge tact multi navigateur

## JENKINS - MODE MAITRE / ESCLAVE

- Le maître
  - défini les esclaves
  - agrège les données
- Les esclaves
  - répartir la charge
  - rendre scalable
  - permet d'avoir des environnements spécifiques (OS, version java, ···)

- Description du job sous forme de code
- Permet la réutilisation
- Historisation du job dans le gestionnaire de source
- Exécution d'étapes en parallèle



```
tools {
   maven 'Maven'
stages{
   stage('Checkout') {
       steps {
           checkout scm: [
               $class: 'GitSCM',
               branches: [[name: '*/master']],
               userRemoteConfigs: [[url: 'file:///home/project/cours_ic']]
   stage('Build and test') {
       steps {
          sh "mvn install"
   stage('Post build') {
       parallel {
           stage('Javadoc') {
              steps {
                  sh "mvn javadoc:javadoc"
                  // publish html
                  publishHTML target: [
                      reportDir: 'target/site/apidocs',
                      reportFiles: 'index.html',
                      reportName: 'Javadoc'
           stage('Checkstyle') {
              steps {
                  sh "mvn checkstyle:checkstyle"
                  publishHTML target: [
```

# QUESTIONS