

INTÉGRATION CONTINUE

Intégration continue S. Fauvel – Univ. Nantes 2022

PLAN

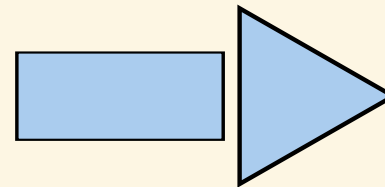
- Intégration
- Intégration continue
- Jenkins

INTÉGRATION

ENTRE LE DÉVELOPPEMENT ET LA PRODUCTION ?

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

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



On appelle "intégration" tout ce qu'il reste à faire à une équipe projet, quand le travail de développement à proprement parler est terminé, pour obtenir un produit exploitable, "prêt à l'emploi".

— Institut Agile

PLANNING

Test unitaires

Compilation

Analyse de code

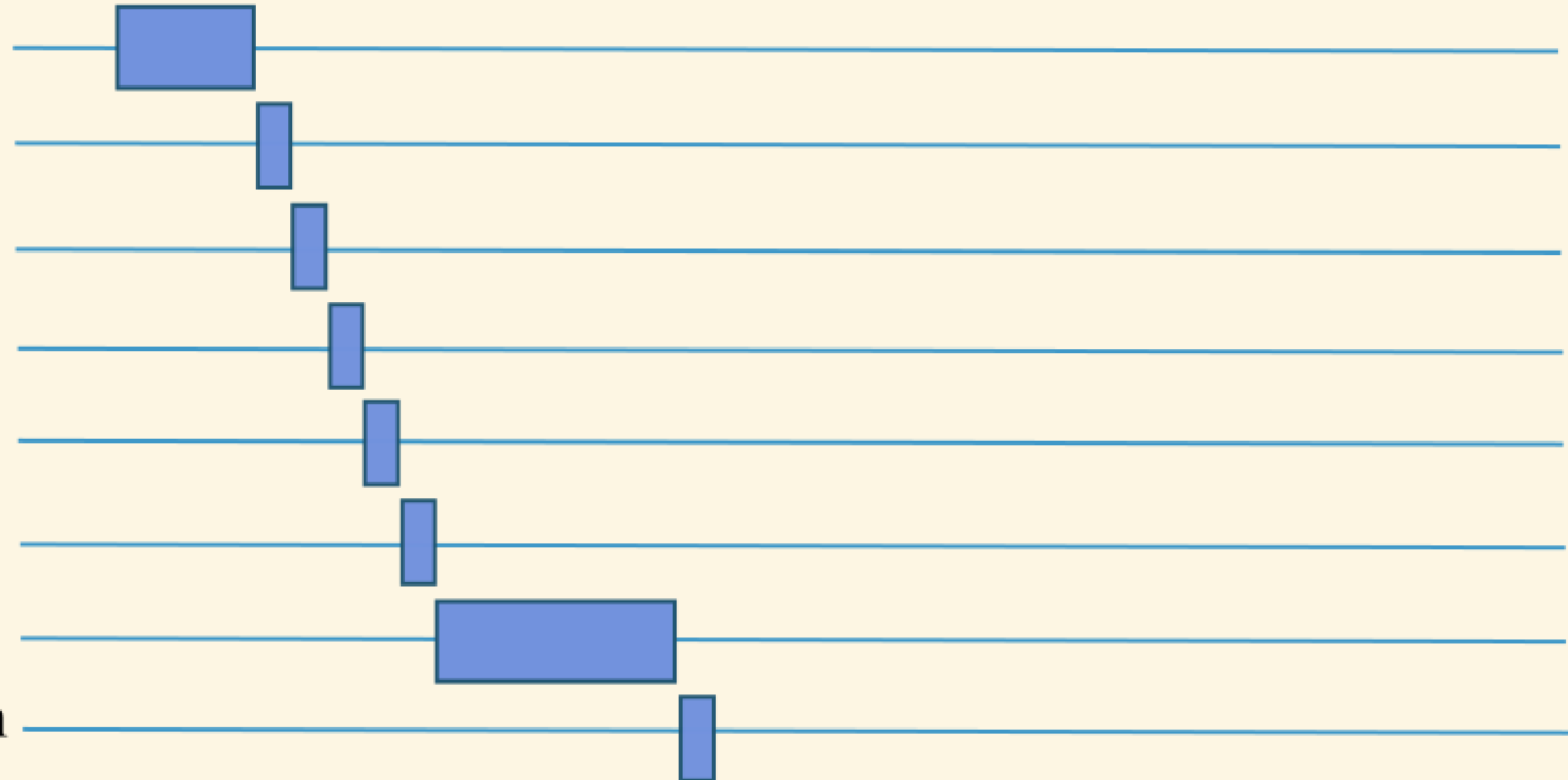
Packaging

Documentation

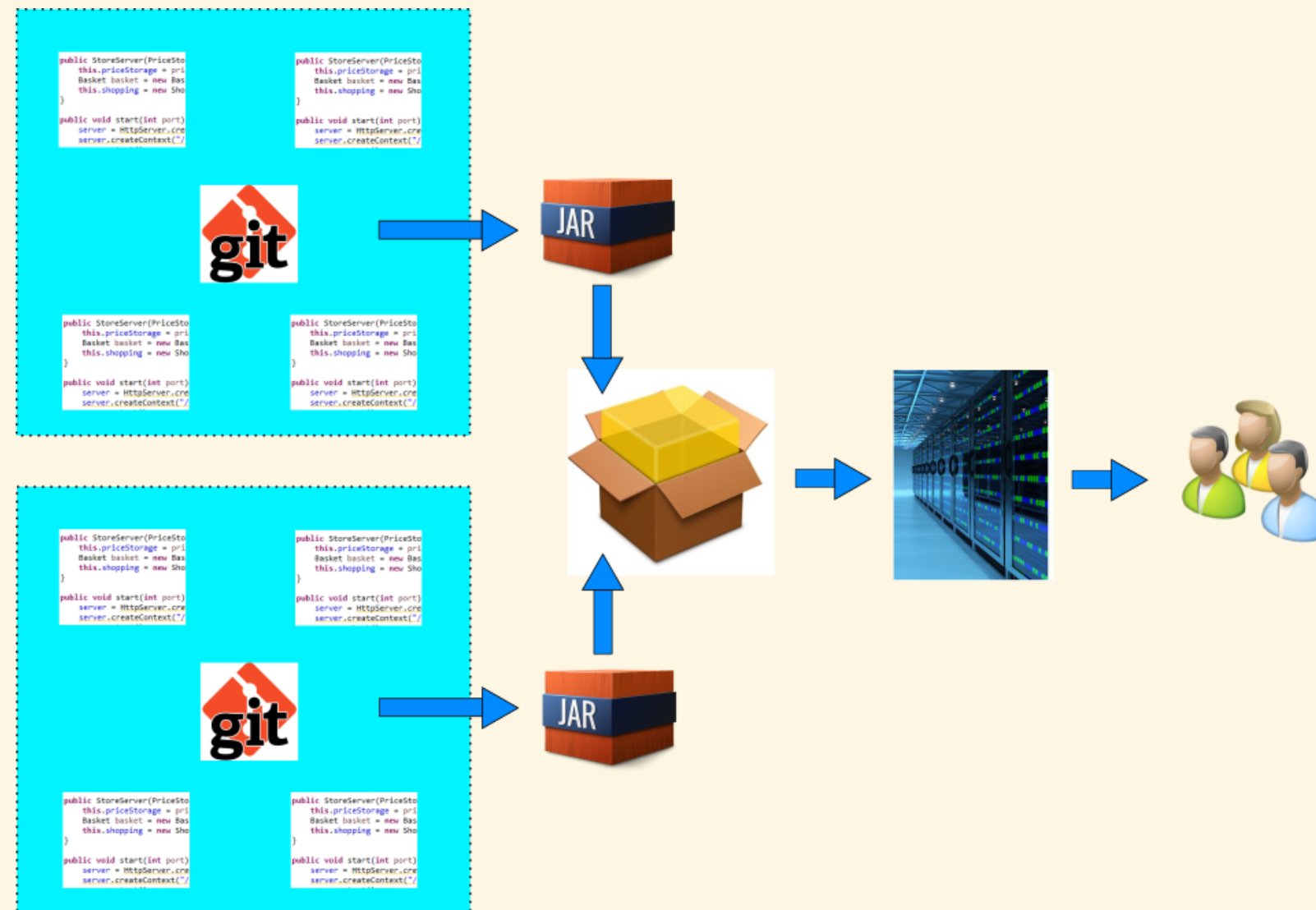
Déploiement

Recette

Mise en production



INTÉGRATION



PLANNING



PLANNING

Test unitaires

Compilation

Analyse de code

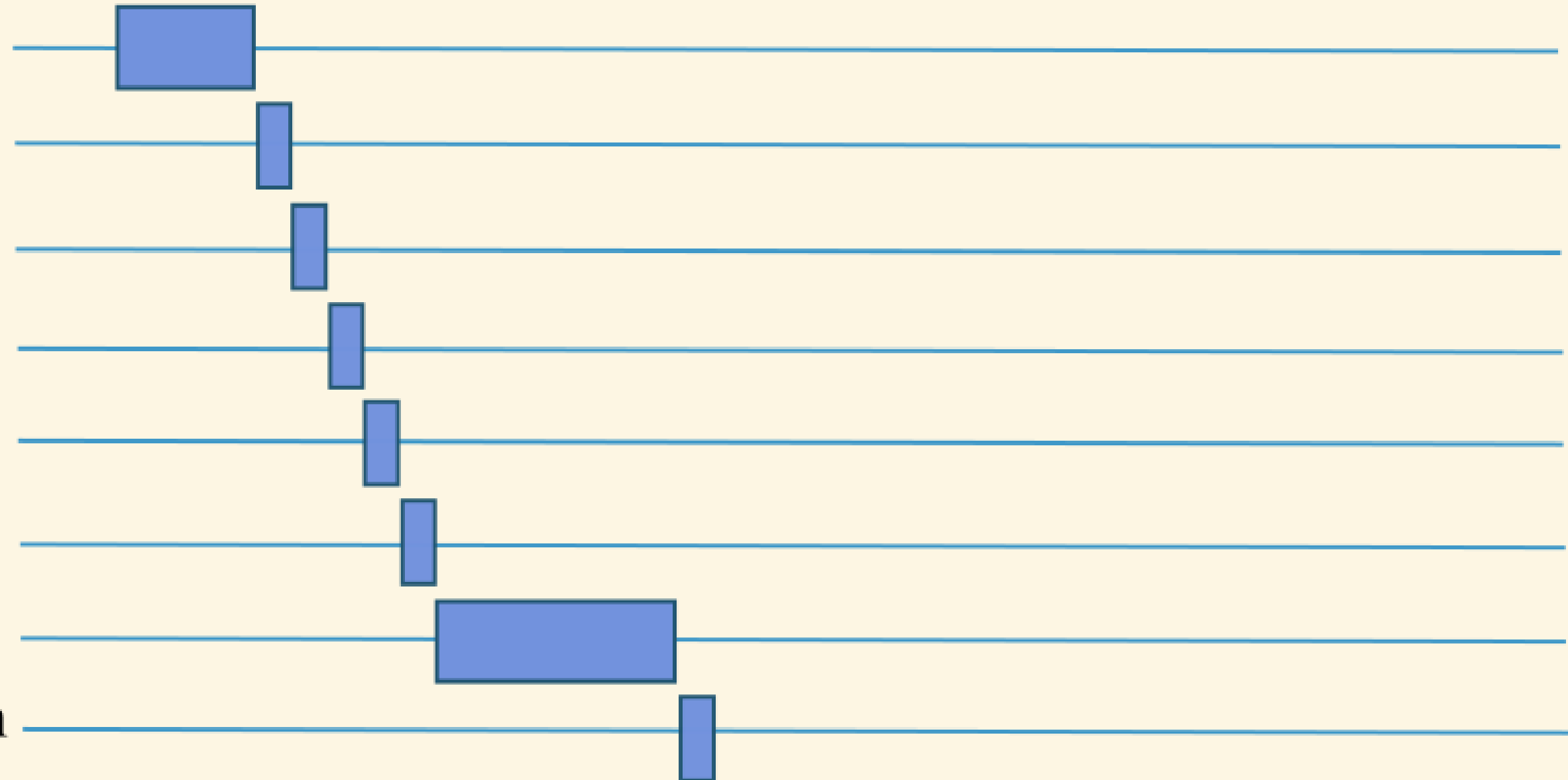
Packaging

Documentation

Déploiement

Recette

Mise en production



EN PRATIQUE

Test unitaires

Compilation

Analyse de code

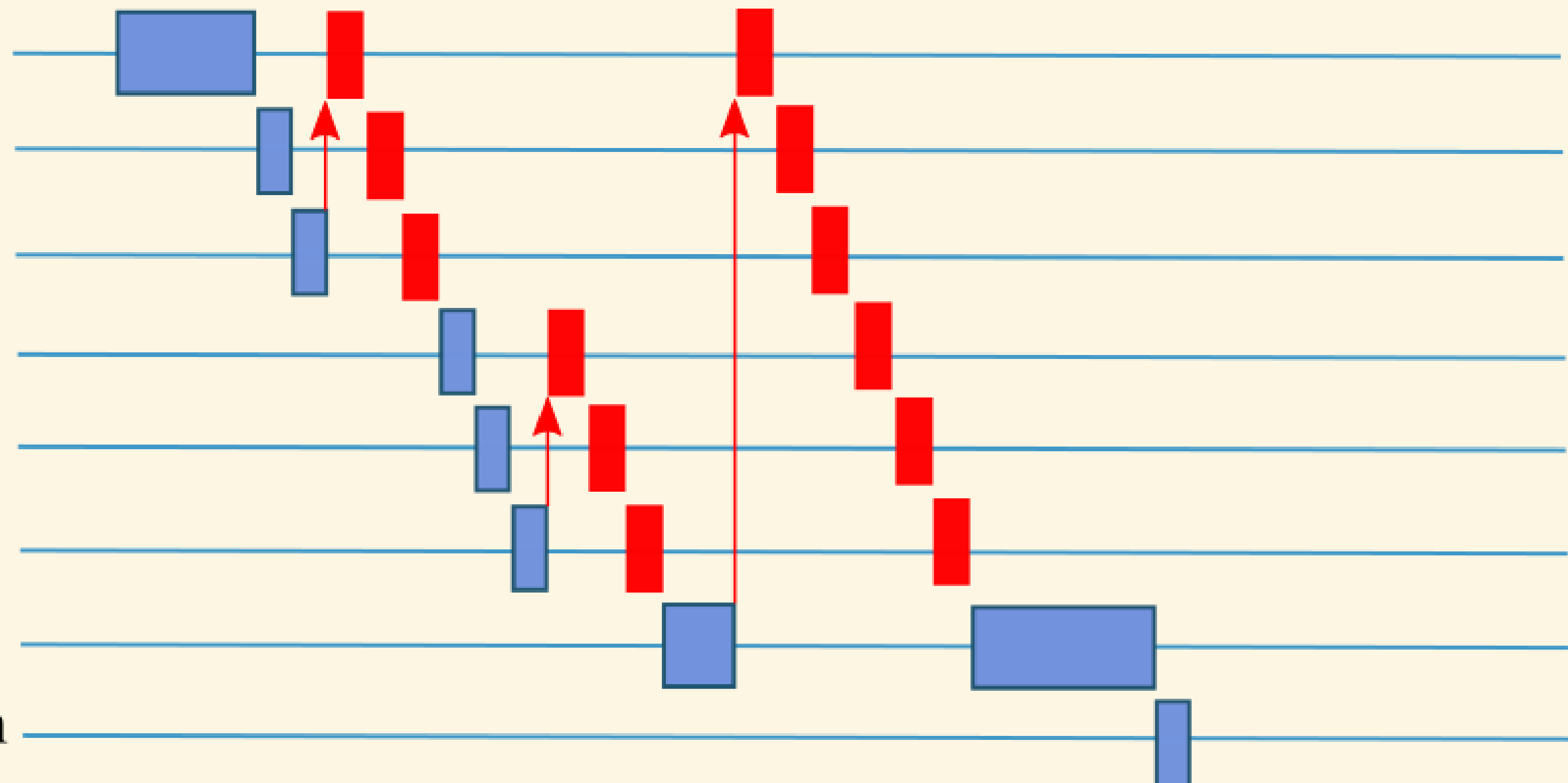
Packaging

Documentation

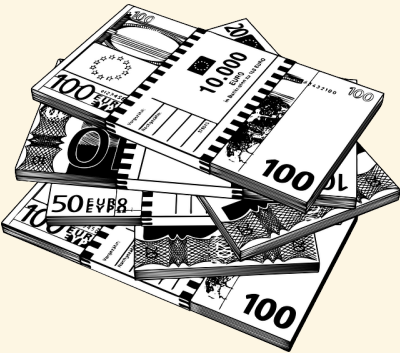
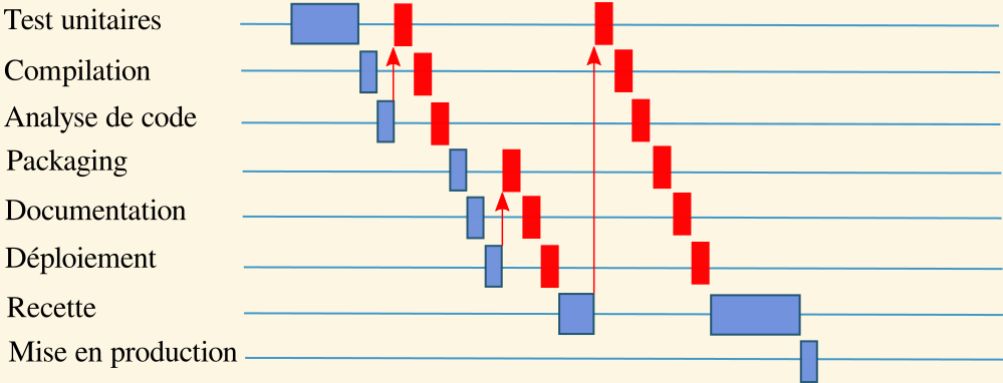
Déploiement

Recette

Mise en production

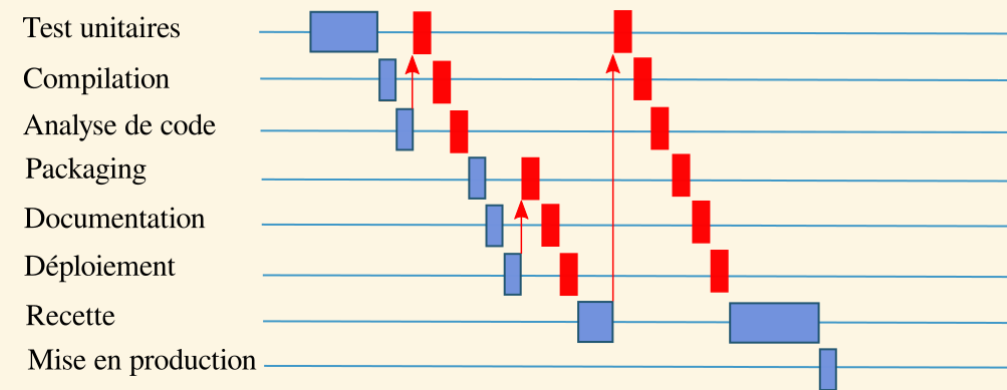


CHARGE



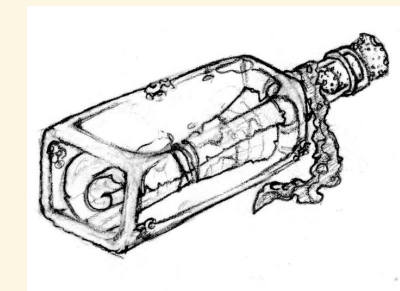
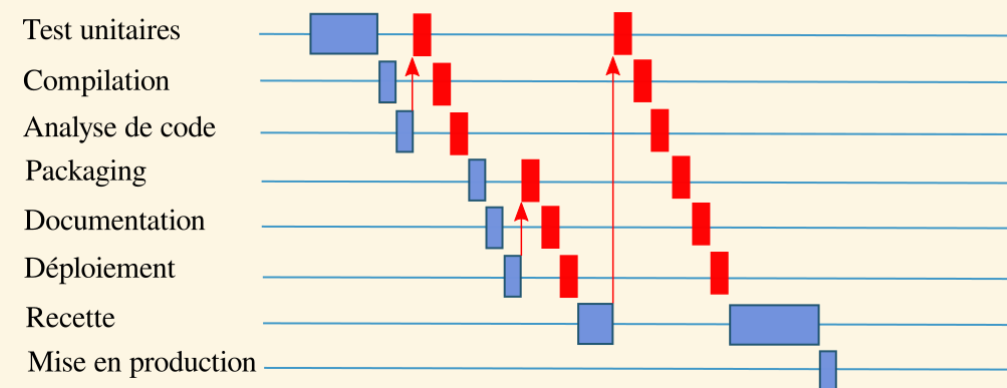
Les étapes à repasser sont une **charge de travail** supplémentaire

TEMPS



Chaque étape à refaire allonge le **délai** de livraison

FEEDBACK



Le temps d'intégration éloigne d'autant le **délai entre le développement et sa validation**

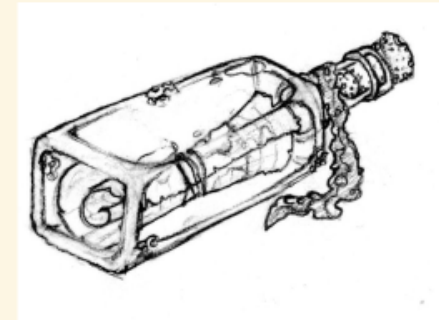
INCERTITUDE



Charges



Temps



Feedback



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 seule 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

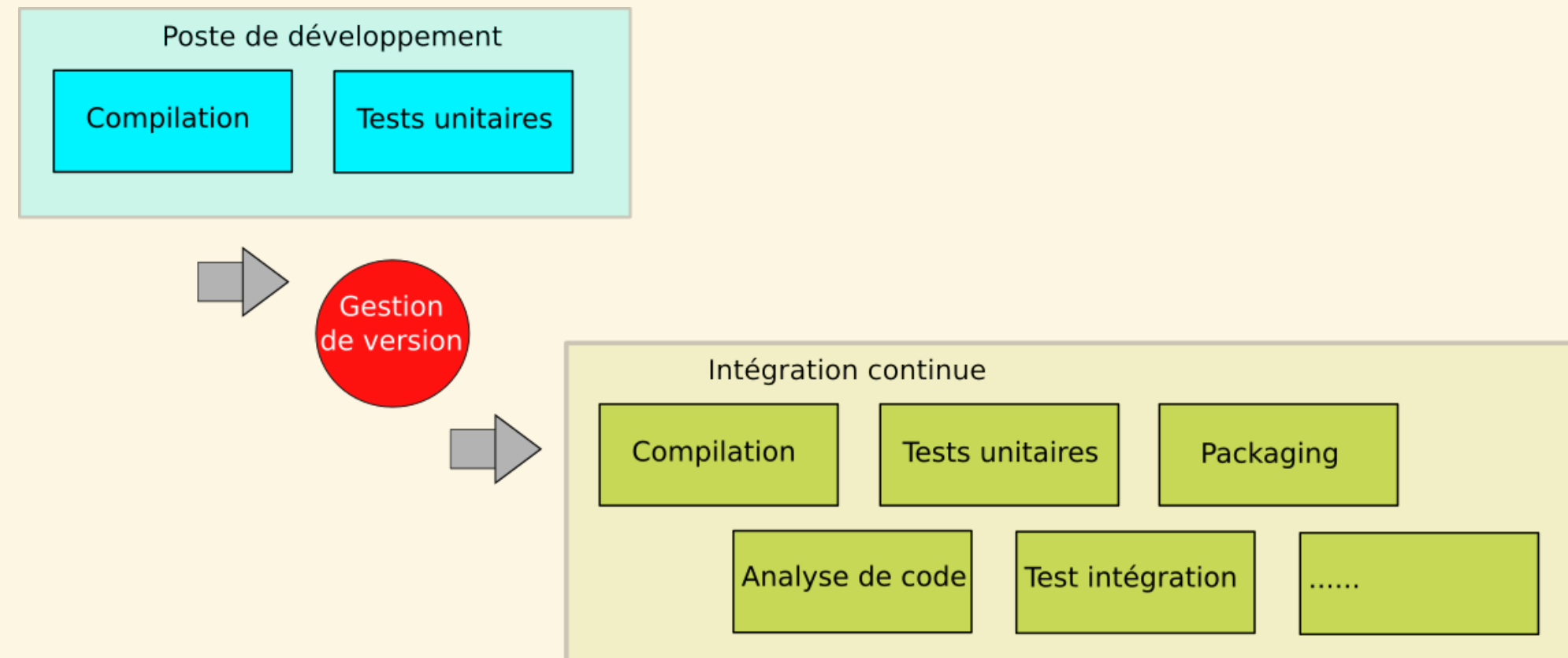
ORIGINE

- 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>

INTÉGRATION CONTINUE

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

BÉNÉFICES

- 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
 - Assure la reproductibilité

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 CI
- GitHub Actions
- ...

JENKINS



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
- ...

Extensible en ajoutant des plugins

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 – TABLEAU DE BORD

Création d'un job

Job

Status du job

Nom

Lancement manuel du job

The screenshot shows the Jenkins dashboard interface. On the left is a sidebar with navigation links: 'Nouveau Item', 'Utilisateurs', 'Historique des constructions', 'Relations entre les builds', 'Vérifier les empreintes numériques', 'Administrer Jenkins', and 'Credentials'. Below these are sections for 'File d'attente des constructions' (empty) and 'État du lanceur de compilations' (showing '1 Au repos' and '2 Au repos'). The main area features a table of jobs. Above the table are buttons for 'Tous' and '+', and a link to 'Ajouter une description'. The table has columns for status (S, M), project name, last success, last failure, and last duration. Each row includes a job icon, a status icon, a link to the job, and a manual launch icon. Annotations with arrows point to specific elements: 'Création d'un job' points to the 'Nouveau Item' link; 'Job' points to the 'SimpleProject' row; 'Status du job' points to the status icon of the 'Once' job; 'Nom' points to the project name 'Epicerie'; and 'Lancement manuel du job' points to the manual launch icon of the 'Once' job.

S	M	Nom du projet ↓	Dernier succès	Dernier échec	Dernière durée
		SimpleProject	10 j - #40	10 j - #46	19 s
		ShUnitTest	1 an. 6 mo. - #8	1 an. 6 mo. - #7	0,36 s
		ProjetSousGit	9 j 23 h - #2	9 j 23 h - #3	9,1 s
		Once	2 an. 2 mo. - #1	s. o.	1 mn 45 s
		Epicerie	1 j 23 h - #1	s. o.	29 s
		BasicProj	22 h - #129	s. o.	0,51 s

Icône: [S](#) [M](#) [L](#)

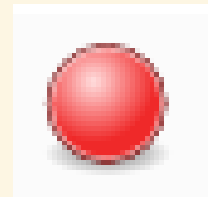
[Légende](#) [RSS pour tout](#) [RSS de tous les échecs](#) [RSS juste pour les dernières compilations](#)

JENKINS – STATUT DU JOB



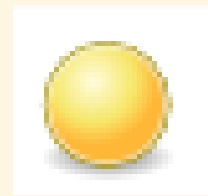
Succès

Tout fonctionne



En échec

La construction n'a pas pu se faire (compilation, ...)



Instable

La build a abouti mais des tests échouent



Désactivé

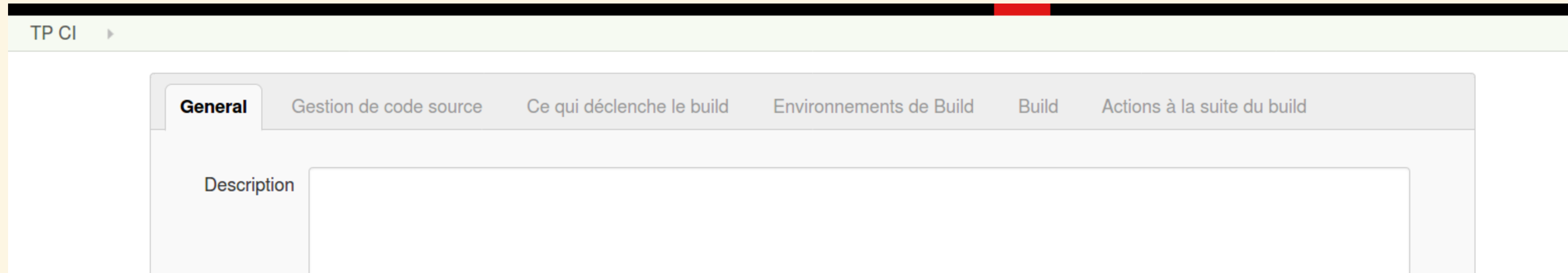
Le job est désactivé

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

JENKINS - JOBS








- Général
- Gestion de code source
- Ce qui déclenche le build
- Environnement de Build
- Build
- Actions à la suite du build

GÉNÉRAL

Description

[Plain text] [Prévisualisation](#)

- ☐ Ce build a des paramètres 
- ☐ GitHub project
- ☐ This build requires lockable resources
- ☐ Supprimer les anciens builds 
- ☐ Throttle builds 
- ☐ Désactiver le projet 
- ☐ Exécuter des builds simultanément si nécessaire 

Avancé...

- Paramètres du build
- Supprimer les anciens build
- ...

GESTION DE CODE SOURCE

Gestion de code source

☐ Aucune

☒ Git

Repositories

Repository URL

/home/project/cours_ic

Credentials

- aucun -

Ajouter

Avancé...

Add Repository

Branches to build

Branch Specifier (blank for 'any')

*/master

Add Branch

Navigateur de la base de code

(Auto)

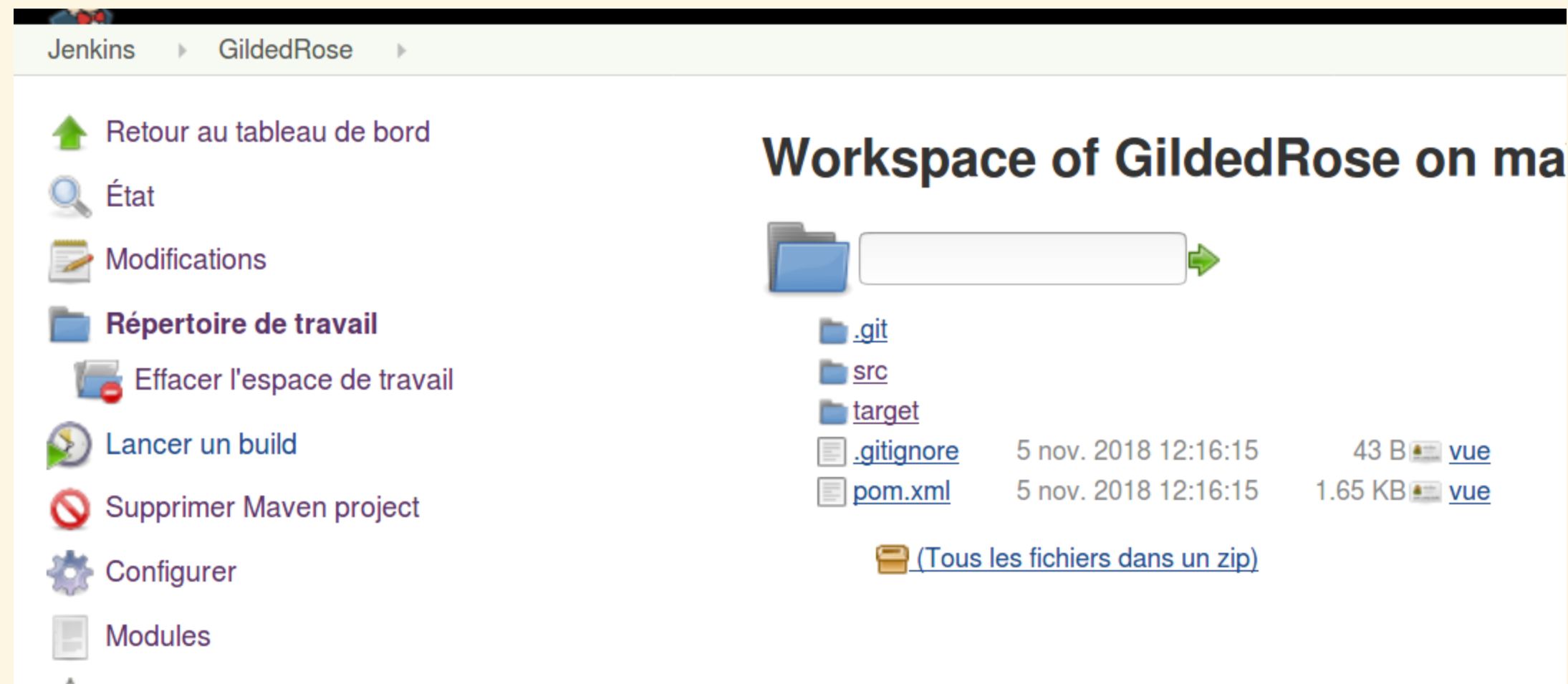
Additional Behaviours

Ajouter

☐ Mercurial

GESTION DE CODE SOURCE

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



CE QUI DÉCLENCE LE BUILD

Ce qui déclenche le build

- ☐ Déclencher les builds à distance (Par exemple, à partir de scripts)
- ☐ Construire après le build sur d'autres projets
- ☐ Construire périodiquement
- ☐ GitHub hook trigger for GITScm polling
- ☒ Scrutation de l'outil de gestion de version



Planning

H/15 * * * *



Aurait été lancé à Sunday, November 4, 2018 8:44:21 PM UTC; prochaine exécution à Sunday, November 4, 2018 8:59:21 PM UTC.

Ignore post-commit hooks





☐

JENKINS - DÉCLENCHEMENT


- 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:
 - toutes les 5 minutes: */5 * * * *
 - A 23h30 le vendredi : 30 23 * * 5


JENKINS – GESTIONNAIRE DE CONFIGURATION


Déclenchement toutes les 2 minutes

 #4	25 oct. 2018 21:23
 #3	25 oct. 2018 21:21
 #2	25 oct. 2018 21:19
 #1	25 oct. 2018 21:17

Dernier accès au gestionnaire de configuration

 Javadoc

 Log du dernier accès à Git

 Open Blue Ocean

Dernier Log du dernier accès à Git

```
Started on Nov 5, 2018 12:28:00 PM
Using strategy: Default
[poll] Last Built Revision: Revision fb3eb3a8c455c03bd4fcd2917db495781282b508 (refs/remotes/origin/master)
> git --version # timeout=10
> git ls-remote -h /home/project/cours_ic # timeout=10
Found 1 remote heads on /home/project/cours_ic
[poll] Latest remote head revision on refs/heads/master is: fb3eb3a8c455c03bd4fcd2917db495781282b508 - already built
by 26
Done. Took 14 ms
No changes
```

ENVIRONNEMENT DE BUILD

Environnements de Build

- ☐ Delete workspace before build starts
- ☐ Use secret text(s) or file(s)
- ☐ Abort the build if it's stuck
- ☐ Add timestamps to the Console Output
- ☐ Generate Release Notes



BUILD

Build

Ajouter une étape au build ▼

Exécuter un script shell

Exécuter une ligne de commande batch Windows

Invoquer les cibles Maven de haut niveau

JIRA : Ajout des variables d'environnement à la construction

JIRA : Appliquer une transition de flux de travaux aux demandes.

JIRA : Créer une nouvelle version

JIRA : Marquer une version réléasée

JIRA : Mise à jour d'un champ personnalisé

Run with timeout

Set build status to "pending" on GitHub commit

BUILD

Script shell

Build

Exécuter un script shell

Commande `echo "Version 1.0" > version.txt`

Voir [la liste des variables d'environnement disponibles](#)

Avancé...

Ajouter une étape au build ▼

Build Maven

Build

Invoquer les cibles Maven de haut niveau

Version de MavenMaven

Cibles Mavenclean install▼

Avancé...

Ajouter une étape au build ▼

JENKINS – INFORMATIONS SUR L’EXÉCUTION

Sur le tableau de bord

MIAGE						Ajouter une description
Tous	+					
S	M	Nom du projet ↓	Dernier succès	Dernier échec	Dernière durée	
		Epicerie	2 j 23 h - #1	s. o.	29 s	
		BasicProj	1 j 0 h - #132	s. o.	0,67 s	

Sur le job

Modifications

Répertoire de travail

Supprimer Projet

Configurer

Historique des builds

[tendance](#) ⇌

#132	14 juin 2015 23:37
#131	14 juin 2015 23:36
#130	14 juin 2015 23:35
#129	14 juin 2015 00:38
#128	14 juin 2015 00:37

Ce projet est actuellement désactivé

Activer

Espace de travail

Changements récents

Liens permanents

- [Dernier build \(#132\), il y a 1 j 0 h](#)
- [Dernier build stable \(#132\), il y a 1 j 0 h](#)
- [Dernier build avec succès \(#132\), il y a 1 j 0 h](#)

Console

Retour au projet

État

Modifications

Console Output

View as plain text

Informations de la construction

Supprimer le build

Build précédent

Sortie de la console

Lancé par une alarme périodique

Building in workspace C:\Users\jrgf2315\.jenkins\workspa

[BasicProj] \$ cmd /c call C:\Users\jrgf2315\AppData\Loca

C:\Users\jrgf2315\.jenkins\workspace\BasicProj>C:\Users\

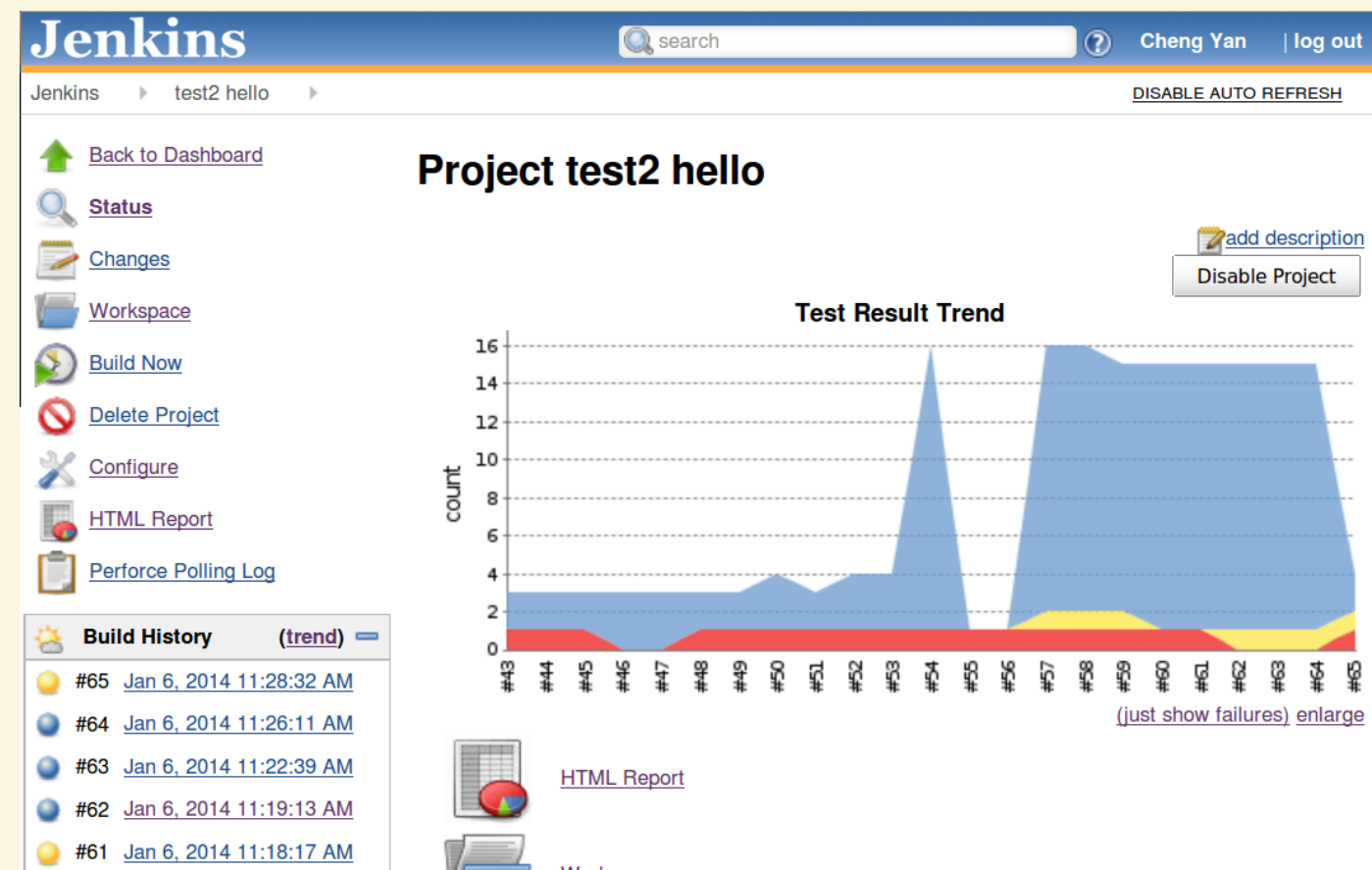
C:\Users\jrgf2315\.jenkins\workspace\BasicProj>set texte

C:\Users\jrgf2315\.jenkins\workspace\BasicProj>echo 14/0

Finished: SUCCESS

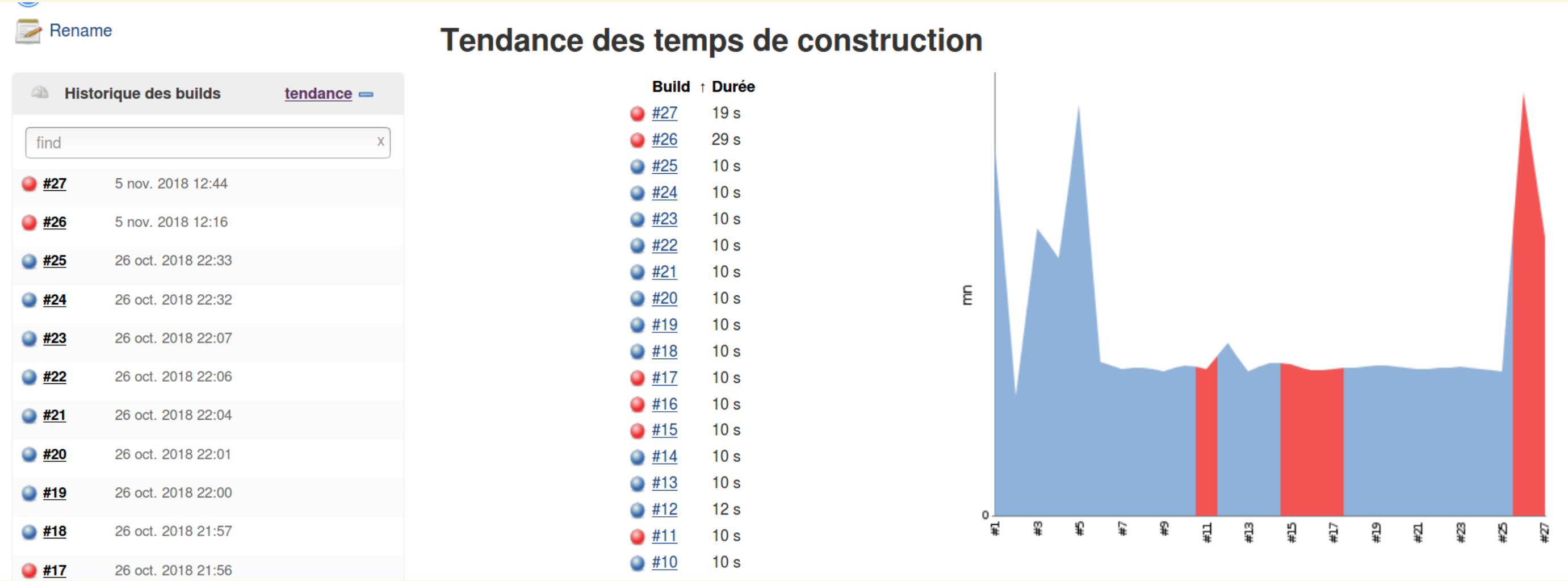
JENKINS – TESTS

- 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



JENKINS – TENDANCE

- Visualisation du temps de construction du build



ACTIONS À LA SUITE DU BUILD

8:50:21 PM UTC

Publier les résultats de l'analyse Checkstyle

Archiver des artefacts

Consolider les résultats des tests en aval

Construire d'autres projets (projets en aval)

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

JIRA : Mise à jour des demandes

Publier le rapport des résultats des tests JUnit

Publier les Javadocs

Publish HTML reports

Git Publisher

JIRA : Créer une demande

JIRA : Créer une nouvelle version

JIRA : Déplacer les demandes correspondant au JQL vers la version spécifiée

JIRA : Marquer une version réléasée

Notifier par email

Set GitHub commit status (universal)

Set build status on GitHub commit [deprecated]

Delete workspace when build is done

Ajouter une action après le build ▼

?

?

JENKINS – NOTIFICATION

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.

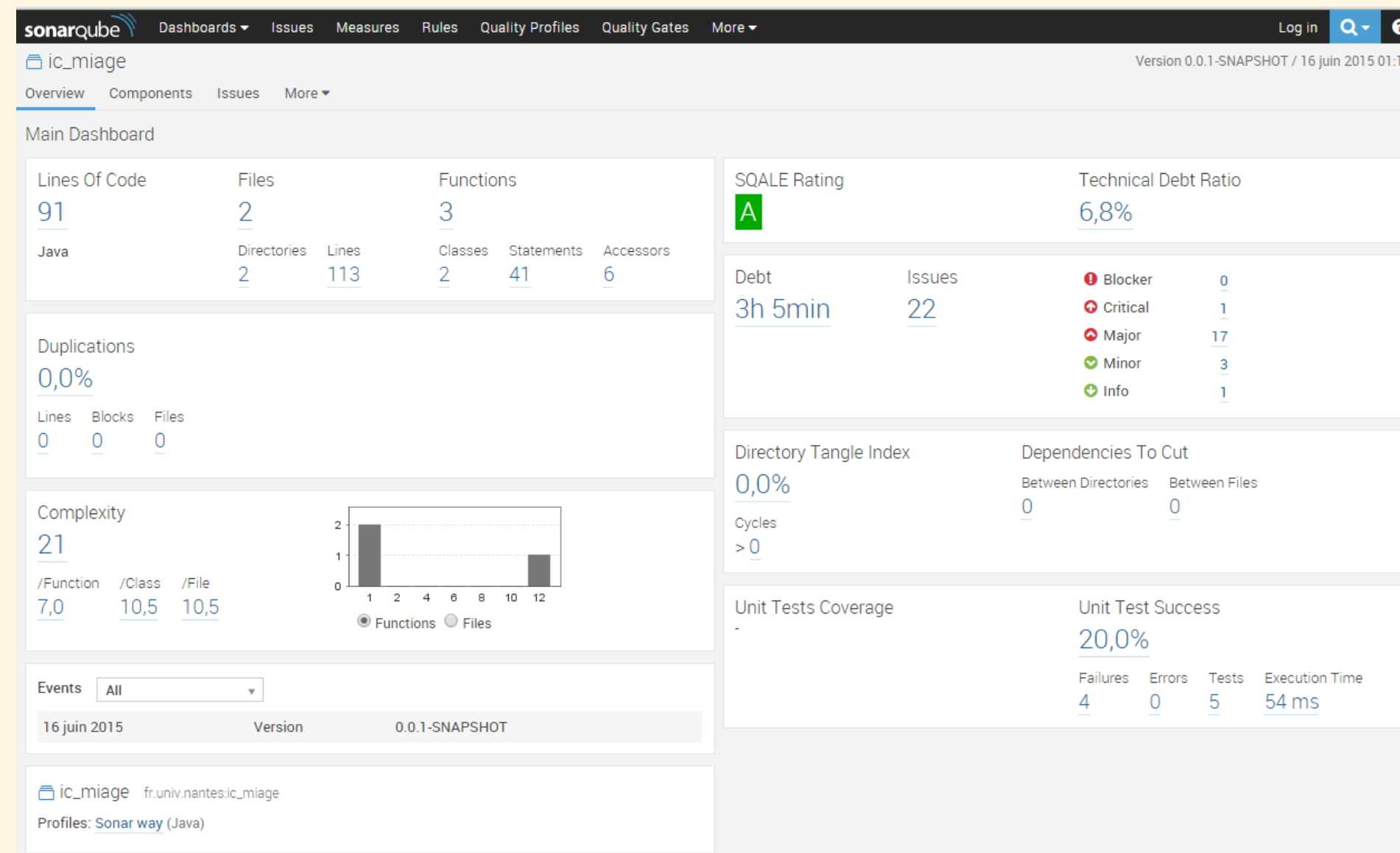
D'autres moyens peuvent être utilisés pour signaler le problème



La notification peut être 'ciblée' sur celui qui a cassé le build (nécessite d'ajouter les utilisateurs)

JENKINS – EXÉCUTION POST BUILD: ANALYSE DE CODE

- D'autres actions peuvent être réalisées 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



JENKINS – ORGANISATION DES BUILDS

- 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
 - Hebdomadaire: test de charge, test multi-navigateur

JENKINS – MODE MAÎTRE / ESCLAVE

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

JENKINS – PIPELINE

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

✓ Pipeline < 43

Pipeline

Modifications

Tests

Artefacts

↺

⚙️

🔗

Déconnexion

✕

Branche: —

🕒 8s

Aucune modification

Commit: —

🕒 2 minutes ago

Démarré par l'utilisateur admin

Start

Checkout

Build and test

Post build

End

✓

✓

✓

Checkstyle

Javadoc

Post build / Javadoc - <1s

🔄 Restart Post build

🔗

⬇️

✓	> Maven — Use a tool from a predefined Tool Installation	<1s
✓	> Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step.	<1s
✓	> mvn javadoc:javadoc — Shell Script	4s
✓	> Publish HTML reports	<1s

JENKINS – PIPELINE

```
pipeline {
  agent any
  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: [
              reportDir: 'target/site',
              reportFiles: 'checkstyle.html',
              reportName: 'Checkstyle'
            ]
          }
        }
      }
    }
  }
}
```

QUESTIONS