

# INTEGRATION CONTINUE

Jenkins, hudson, ...

Olivier Barais,



# Jenkins

# INTRODUCTION

- Qu'est ce que l'intégration continue ?
- Pourquoi automatiser ?
- Par où commencer ?
- Le cycle vertueux de l'intégration continue

# Définitions

***"L'intégration continue est un ensemble de pratiques utilisées en génie logiciel. Elles consistent à vérifier à chaque modification de code source que le résultat des modifications ne produit pas de régression de l'application en cours de développement."***

Wikipedia

***"Une pratique considérant différemment l'intégration, habituellement connue comme pénible et peu fréquente, pour en faire une tâche simple faisant partie intégrante de l'activité quotidienne d'un développeur."***

Documentation  
CruiseControl.NET

# Qu'est ce que l'intégration continue ?

- Technique puissante permettant dans le cadre du développement d'un logiciel en équipes de:
  - Garder en phase les équipes de dev
  - Limiter risques de dérive
  - Limiter la complexité
- A intervalles réguliers, vous allez construire (build) et tester la dernière version de votre logiciel.
- Parallèlement, chaque développeur teste et valide (commit) son travail en ajoutant son code dans un lieu de stockage unique.

# Pourquoi automatiser ?

- Gagner du temps
  - Vous ne faites pas de tâches répétitives
- Gagner en confiance
  - Indépendant de votre efficacité du moment
  - Procédures répétables
- Diminue le besoin de documentation
  - Pour nouveaux entrants projet, utiliser scripts !
  - ...et + en analysant le script.

# Par où commencer ?

- 1) Outil gestion versions code sources
  - Lieu unique de partage
  - Retour arrières, snapshots, branches...

- 2) Tests automatisés
  - Chaque développeur



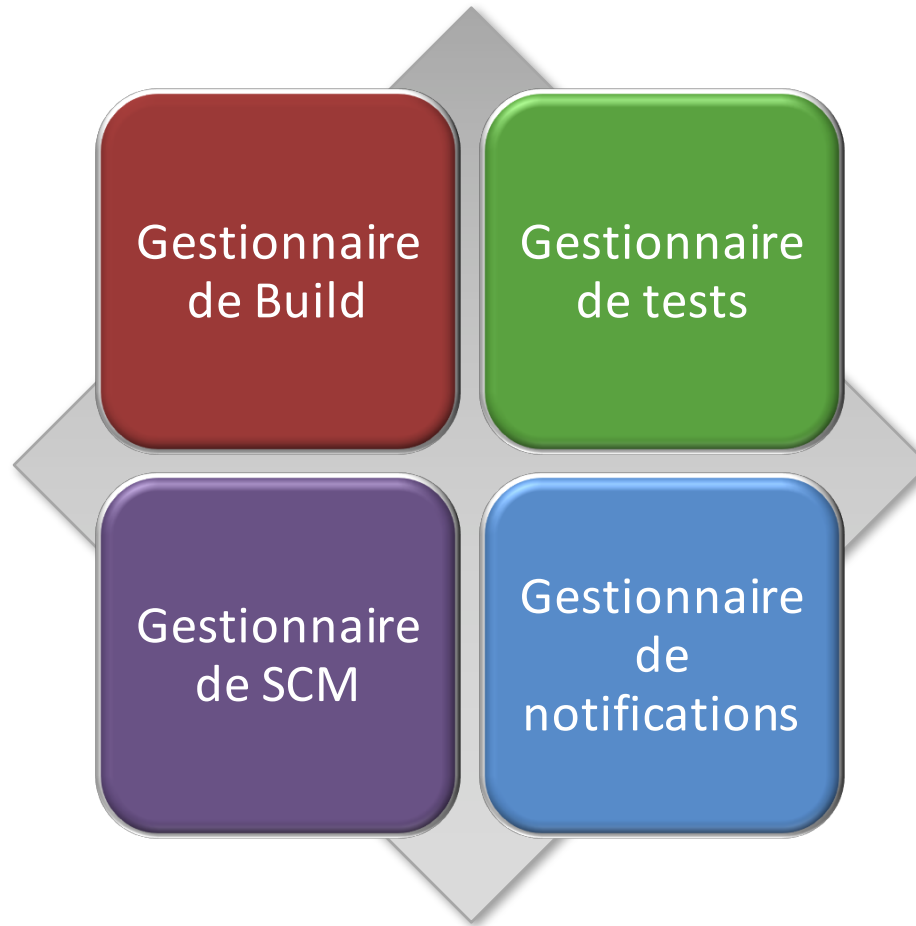
- 3) Scripts
  - Coté serveur pour automatiser (Ex : crontab)



- 4) Outils de communication
  - Mail, Tél, Rss...



# Architecture d'un logiciel d'intégration



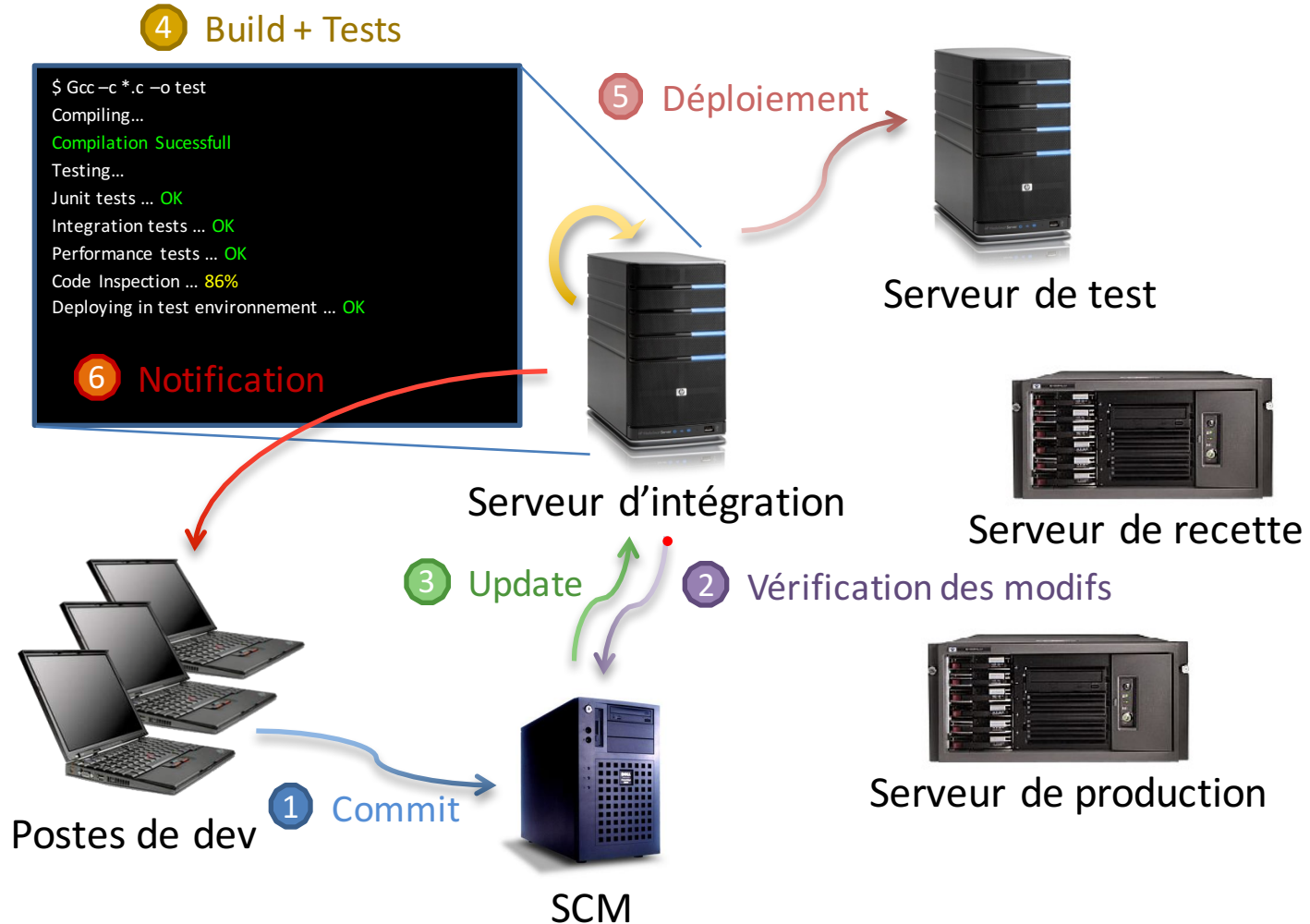
# Un fonctionnement actif

- ➡ Les développeurs « committent »
- ➡ Le serveur d'intégration surveille le serveur SCM (Cron)



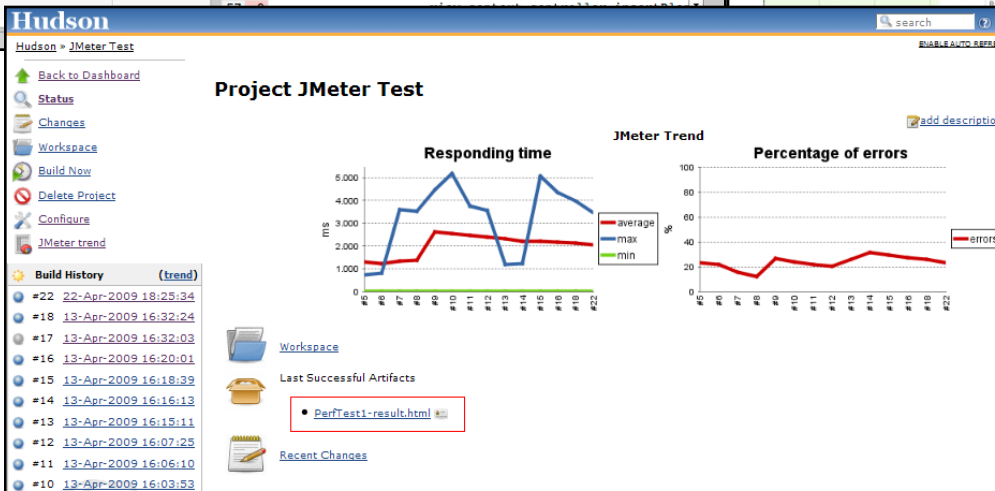
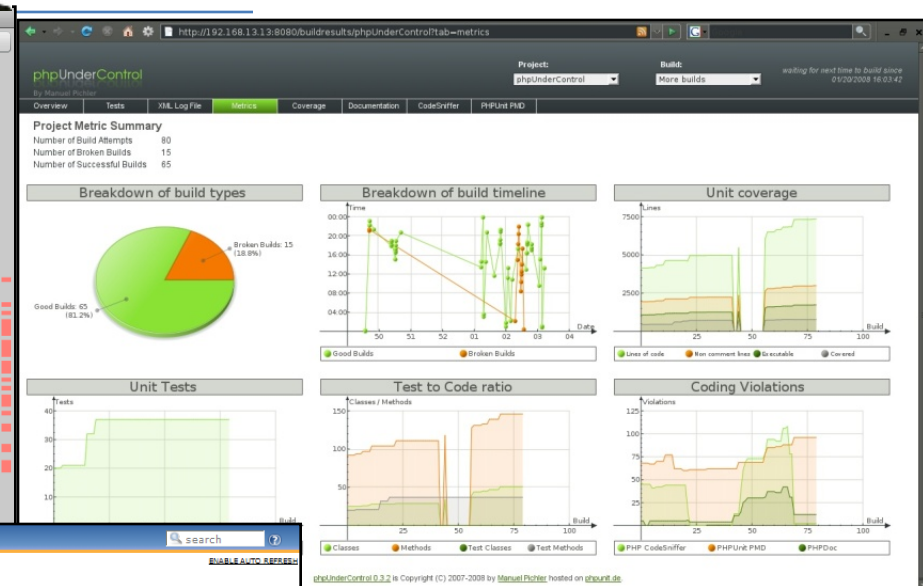
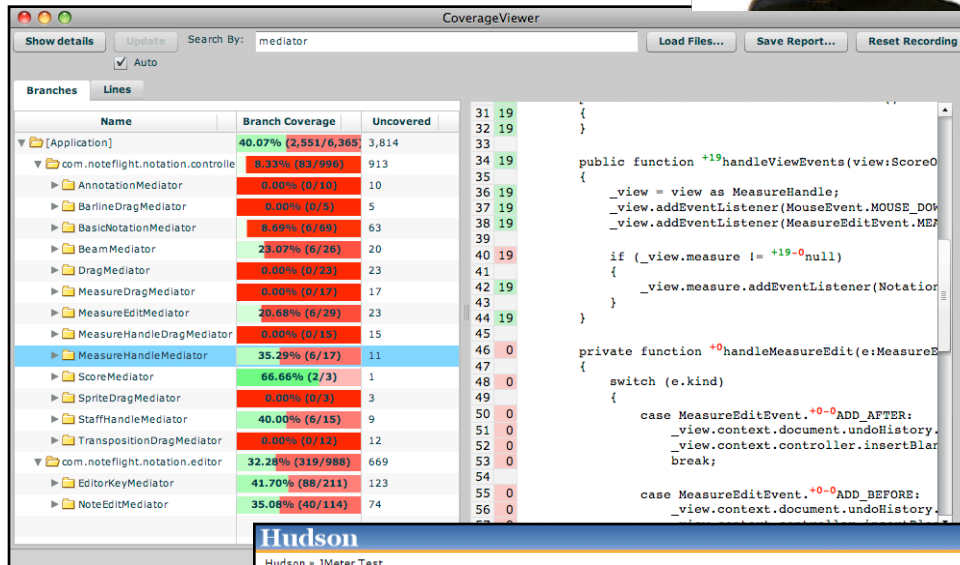
# Cas d'utilisation

## Le développeur soumet une modification



# Cas d'utilisation

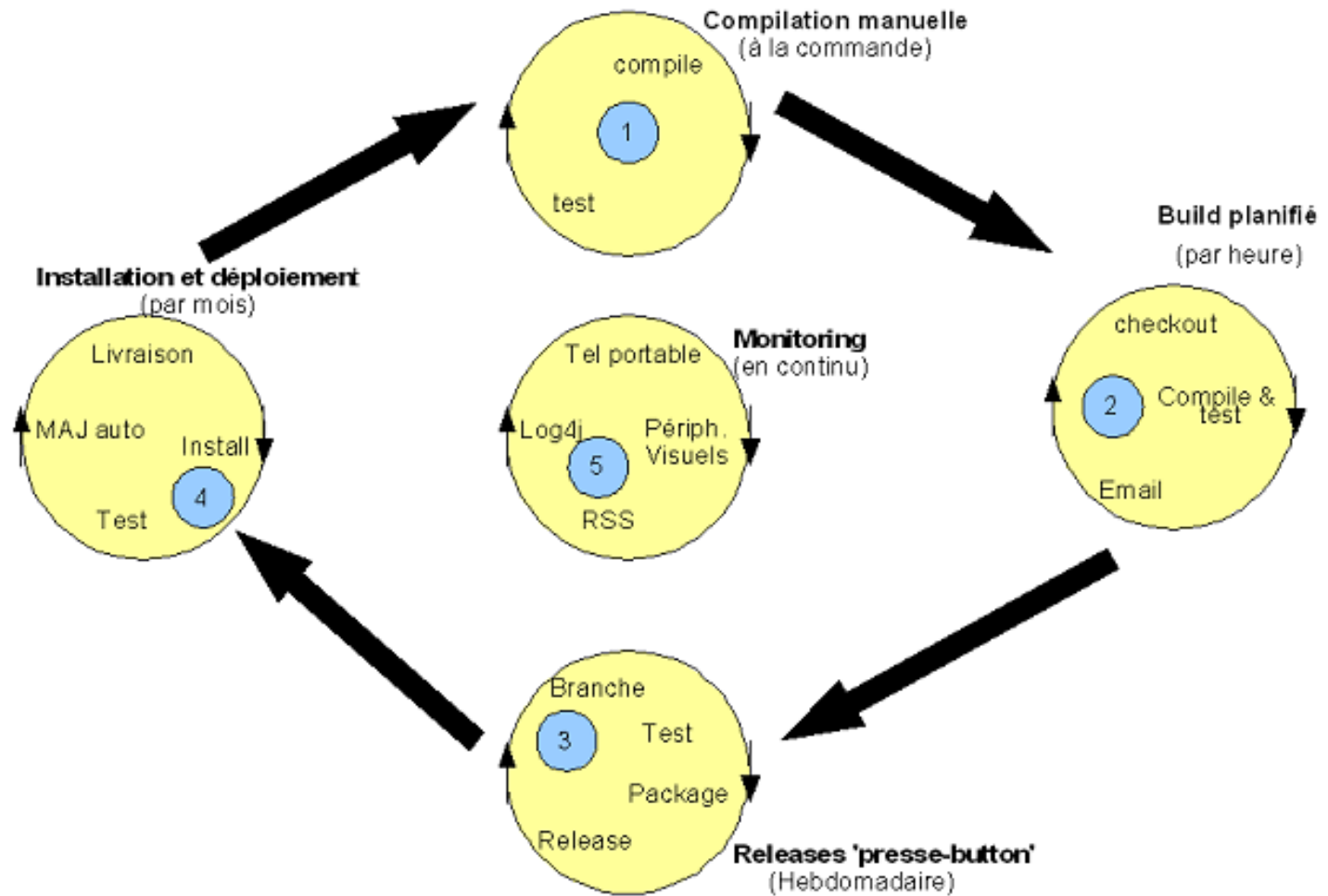
## Le chef de projet analyse le reporting



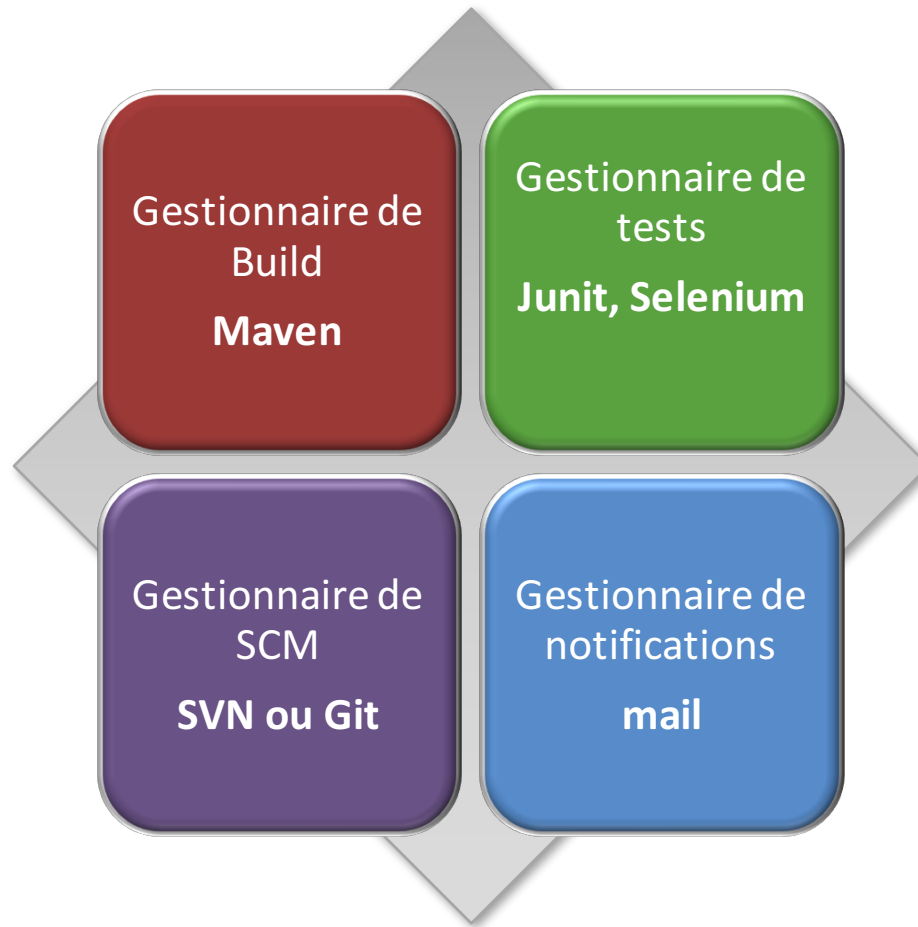
# Les technologies existantes

- ➔ Hudson, jenkins
- ➔ CruiseControl / CruiseControl.NET
- ➔ Apache Continuum
- ➔ QuickBuild (open-source: LuntBuild)
- ➔ Et beaucoup d'autres ...

# Cycle vertueux intégration continue



# Merci



# Questions

