

# Introduction à l'Intégration Continue

Johann Mortara

IUT Nice Côte d'Azur

Département Informatique

20 Septembre 2019

# Rappels sur les tests et la qualité du code

**Règle n°1** : Les tests, c'est TRÈS important !

**Règle n°2** : Écrire des tests, c'est bien. Les exécuter, c'est mieux.

# Ce qu'on vous a appris :

1. Coder
2. Écrire les tests
3. S'assurer que les tests passent
4. Pousser le code sur Git



**Attends c'est  
pas ça qu'il faut  
faire ?**

**Je vois pas où est  
le problème...**

**Je savais bien  
que ça servait à  
rien tout ça !**

**M'aurait-on  
donc menti ?**

**C'est une bonne pratique !**

# Tester, à quoi ça sert ?

# Tester, à quoi ça sert ?

1. Vérifier que le code qui vient d'être écrit a le comportement attendu...

# Tester, à quoi ça sert ?

1. Vérifier que le code qui vient d'être écrit a le comportement attendu...
2. ... et qu'il n'a pas altéré le reste du projet !

# Tester, à quoi ça sert ?

1. Vérifier que le code qui vient d'être écrit a le comportement attendu...
2. ... et qu'il n'a pas altéré le reste du projet !

⇒ maîtriser le code **et son évolution**

# Ce qu'on vous a appris :

1. Coder
2. Écrire les tests
3. S'assurer que les tests passent
4. Pousser le code sur Git



# Ce qu'on vous a appris (v2) :

1. Coder
2. Écrire les tests
3. S'assurer que les tests passent
4. Pousser le code sur Git

## Maîtrise de l'évolution

5. S'assurer que **tous** les tests passent toujours



## Maîtrise du code

# Ce qu'on vous a appris (v2) :

1. Coder
2. Écrire les tests
3. S'assurer que les tests passent
4. Pousser le code sur Git

## Maîtrise de l'évolution

5. S'assurer que **tous** les tests passent toujours

⇒ **Non-régression**



## Maîtrise du code

# Question :

Combien de temps faut-il pour exécuter tous les tests d'un projet ?

- A. < 5 minutes
- B. < 30 minutes
- C. Plusieurs heures
- D. Ça dépend



# Mozilla Firefox

⚙ Settings | 🚩 Report Duplicate



Very High Activity

13,404

I Use This!

🕒 Analyzed 6 days ago. based on code collected 11 days ago.

## Languages

Total Lines :	28,730,555	Code Lines :	20,548,088	Percent Code Lines :	71.5%
Number of Languages :	48	Total Comment Lines :	4,575,107	Percent Comment Lines :	15.9%
		Total Blank Lines :	3,607,360	Percent Blank Lines :	12.6%



Analyzed 6 days ago. based on code collected 11 days ago.

## Languages

Total Lines :	28,730,555	Code Lines :	20,548,088	Percent Code Lines :	71.5%
Number of Languages :	48	Total Comment Lines :	4,575,107	Percent Comment Lines :	15.9%
		Total Blank Lines :	3,607,360	Percent Blank Lines :	12.6%



# Mozilla Firefox

Settings | Report Duplicate



Very High Activity

13,404

I Use This!

Analyzed 6 days ago. based on code collected 11 days ago.

## Languages

Total Lines :	28,730,555	Code Lines :	20,548,088	Percent Code Lines :	71.5%
Number of Languages :	48	Total Comment Lines :	4,575,107	Percent Comment Lines :	15.9%
		Total Blank Lines :	3,607,360	Percent Blank Lines :	12.6%

# Tout doit être testé !

# Firefox, c'est...

- De l'affichage
- Des fichiers de configuration
- Une base de données embarquée
- Un support pour des plugins personnalisés
- La gestion de plusieurs protocoles
- Support de fichiers PDF, MP3, etc...

Et bien d'autres choses encore !

# Comment s'assurer que tout ça fonctionne ?

Comment s'assurer que tout ça fonctionne ?

**Des tests unitaires !**

Comment s'assurer que tout ça fonctionne ?

**Des tests unitaires !**

**C'est tout ?**

# De nouveaux types de test

Besoin de tester les briques séparément ET ensemble

**⇒ tests d'intégration**

# De nouveaux types de test

Besoin de tester les briques séparément ET ensemble

⇒ **tests d'intégration**

Besoin d'essayer le déploiement sur plusieurs systèmes

⇒ **tests opérationnels**

## 4 Testing levels

- 4.1 Unit testing
- 4.2 Integration testing
- 4.3 System testing
- 4.4 Operational acceptance testing

## 5 Testing types, techniques and tactics

- 5.1 Installation testing
- 5.2 Compatibility testing
- 5.3 Smoke and sanity testing
- 5.4 Regression testing
- 5.5 Acceptance testing
- 5.6 Alpha testing
- 5.7 Beta testing

## 5.8 Functional vs non-functional testing

- 5.9 Continuous testing
- 5.10 Destructive testing
- 5.11 Software performance testing
- 5.12 Usability testing
- 5.13 Accessibility testing
- 5.14 Security testing
- 5.15 Internationalization and localization
- 5.16 Development testing
- 5.17 A/B testing
- 5.18 Concurrent testing
- 5.19 Conformance testing or type testing
- 5.20 Output comparison testing

6021 Successful Jobs

**Temps total d'exécution des  
tests de Firefox :**

**950 h  $\approx$  40 jours**

# Réponse :

Combien de temps faut-il pour exécuter tous les tests d'un projet ?

~~A. ← 5 minutes~~

~~B. ← 30 minutes~~

~~C. ← Plusieurs heures~~

D. Ça dépend

... et ça peut être énorme !

# Question :

**Une nouvelle version de Firefox est déployée...**

- A. toutes les 12 heures**
- B. toutes les 3 semaines**
- C. tous les mois**
- D. tous les 2 mois**

## ^ Release timeline

---

Firefox is released at **intervals of six to eight weeks** (not counting urgent patch updates), meaning that every six to eight weeks there will be a new version of Firefox Release.

### **From mozilla-central to mozilla-release**

- Firefox Nightly is released **every 12 hours** with all the changes landed on mozilla-central.
- Every 6 to 8 weeks, we merge the code from mozilla-central to our mozilla-beta branch. The mozilla-beta branch should now only get patches aimed at stabilizing the release. Any patch on mozilla-central that we want backported to our mozilla-beta branch should follow the [approval rules for uplifts](#).
- Beta 1 and Beta 2 are built from this beta branch and used to build and ship Firefox Developer Edition as a stabilization step before shipping Firefox Beta to our much wider Beta audience.
- Starting with Beta 3, Firefox Beta is released twice a week for Desktop, leaving us with 12 to 16 betas every cycle unless we have chemspills leading to additional betas. Firefox Beta 3 is shipped to a subset of our Beta population. The full Beta population gets updated starting with beta 4 only.
- At the end of the Beta cycle, a final build is validated by our QA and tagged for release into the mozilla-release branch.

6 to 8 weeks

every 12 hours

# Réponse :

Une nouvelle version de Firefox est déployée...

- A. toutes les 12 heures
- ~~B. toutes les 3 semaines~~
- ~~C. tous les mois~~
- D. tous les 2 mois

... et c'est testé !

# Comment est-ce possible ?

treeherder

</> Revision `6cbf1430a66e` — Author `btara@mozilla.com`

19 Other Failed Tests

0 Infra Tests

54 Intermittent Tests

6021 Successful Jobs

0 Running Jobs

0 Pending Jobs

6021 Successful Jobs

# L'intégration continue

C'est pas trop tôt !

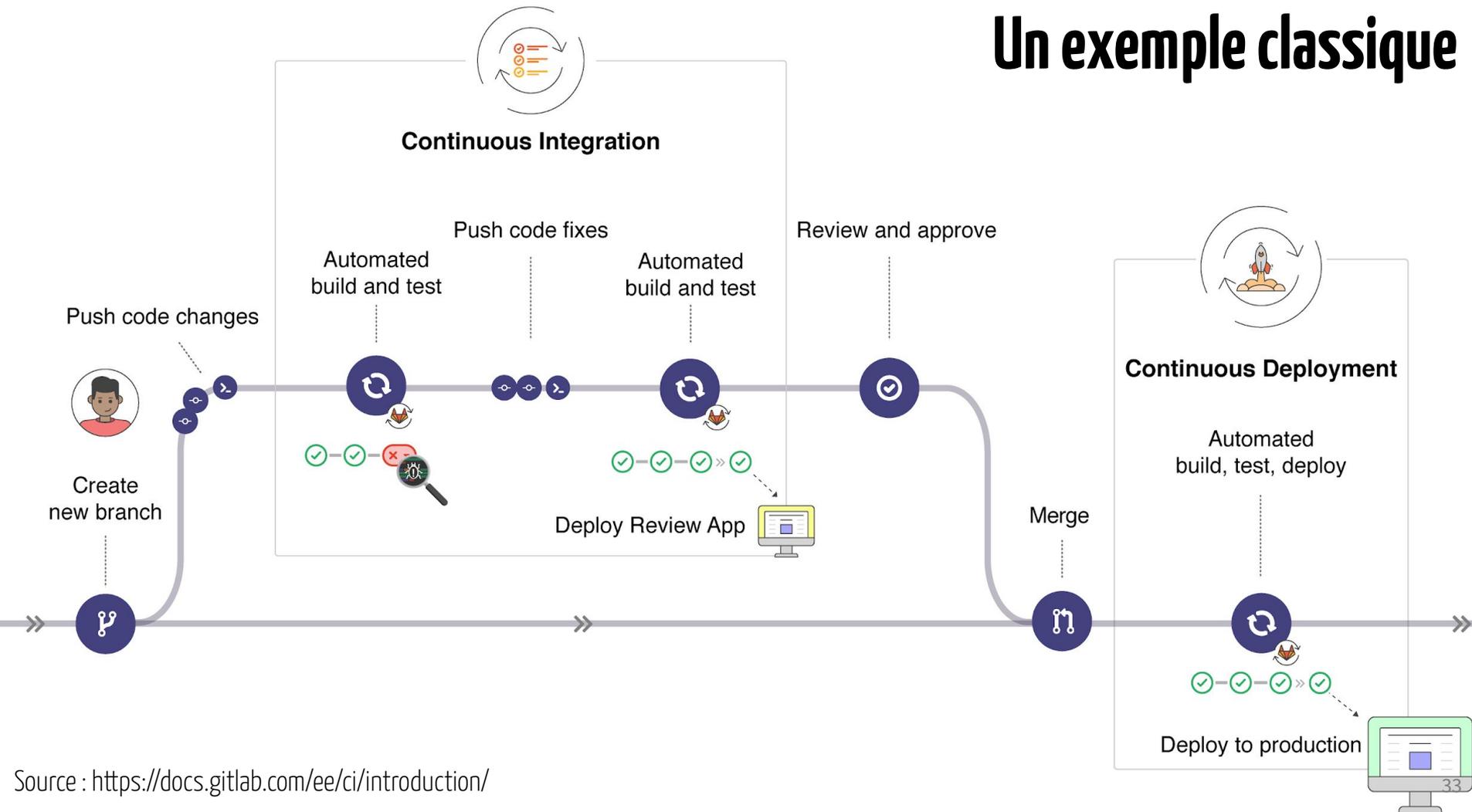
Principe de base

**Automatiser la construction**

**du logiciel pour pouvoir assurer la livraison**

**d'un système qui fonctionne**

# Un exemple classique



# Petit exemple d'application

On conçoit un client de messagerie instantanée (ex. Discord).

**Quels types de tests pourrait-on implémenter ?**

# Tests

Tests de performance (temps au lancement, réactivité de l'application)

Tests de sécurité (chiffrement des échanges)

Tests UI / UX (déclenchement des notifications, affichage du statut des contacts)

Tests d'accessibilité (passage en contraste inversé, tailles des polices)

Tests d'internationalisation (vérification de l'application des traductions)

# Build

	<b>Windows</b>	<b>OSX</b>	<b>Debian</b>	<b>Arch</b>
<b>stable</b>	✓	✓	✓	✓
<b>latest</b>	✓	✓	✓	✓
<b>canary</b>	✓	✗	✓	

Parés au  
déploiement !



# Idées de phases de déploiement

- Création des différents exécutables pour chaque OS (Windows, OSX, Linux...)
- Déploiement des nouvelles versions sur les serveurs
- Mise à jour du site web
- Déclenchement des mises à jour automatiques  
ou notification qu'une mise à jour est disponible

# Des outils pour réaliser de l'intégration continue



**Jenkins**



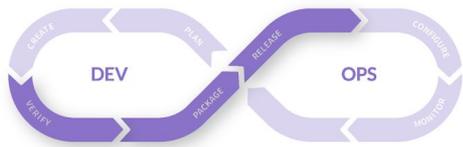
**Travis CI**



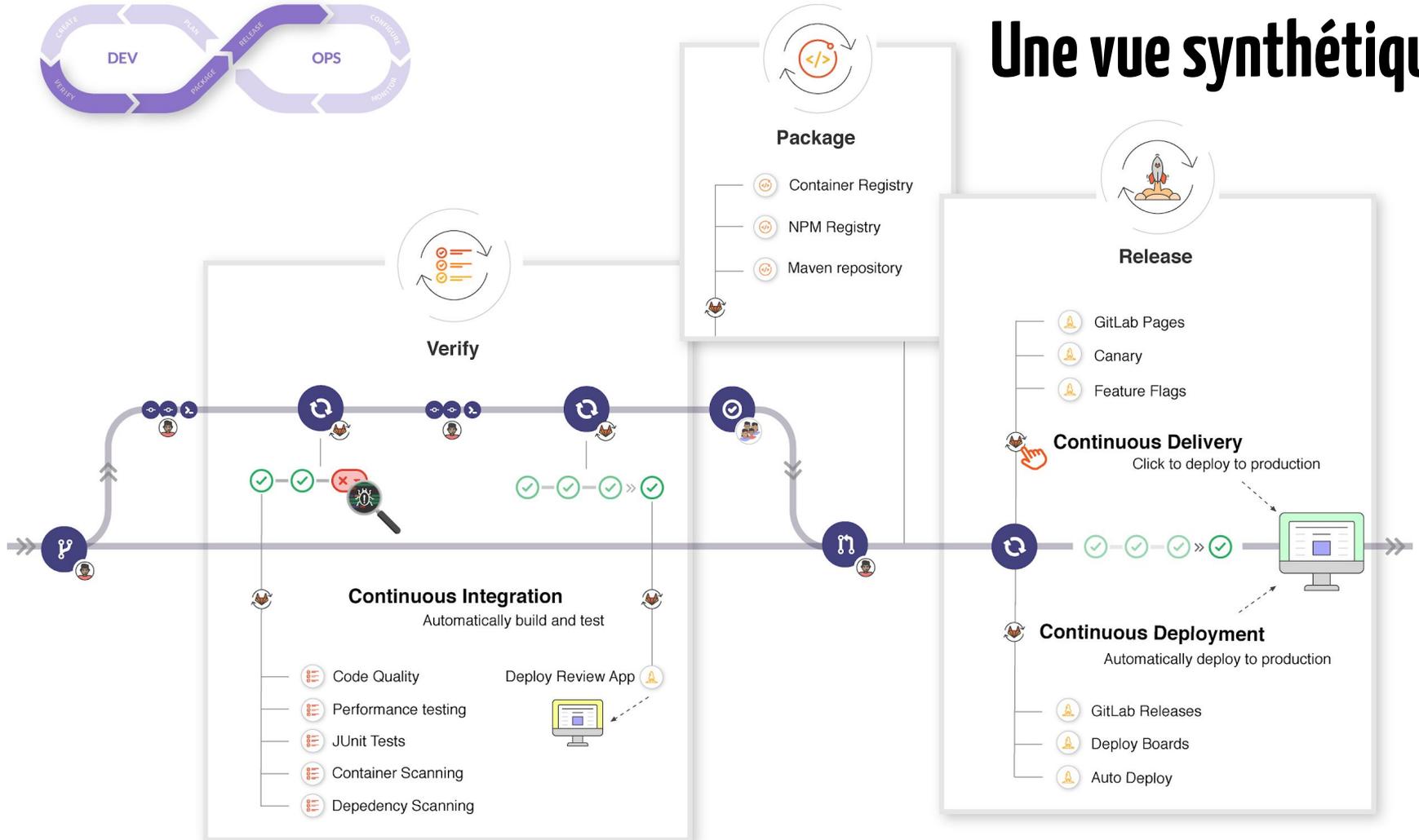
**GitLab CI**

Open Source Continuous Integration made easy





# Une vue synthétique



**Merci !**