Table des matières

[I. Intégration continue 3](#_Toc426446881)

[1. Définition 3](#_Toc426446882)

[2. Pratiques de l'intégration continue 3](#_Toc426446883)

[ Maintenir un gestionnaire de sources unique 3](#_Toc426446884)

[ Automatiser la construction (Build) 3](#_Toc426446885)

[ Rendre le Build « auto-testant » pour capturer les bugs 3](#_Toc426446886)

[ Tout le monde soumet son travail (commit) au même endroit. 4](#_Toc426446887)

# Intégration continue

## Définition

L'intégration continue est une pratique de développement logiciel où les membres d'une équipe intègrent leur travail fréquemment.

En général, chacun intègre au moins quotidiennement - conduisant à de multiples intégrations par jour. Chaque intégration est vérifiée par un système automatisé de construction (y compris les tests) pour détecter les erreurs d'intégration le plus rapidement possible.

Beaucoup d'équipes trouvent que cette approche conduit à une réduction considérable des problèmes d'intégration et permet à une équipe de développer un logiciel de qualité plus rapidement. « Martin Fowler »

## Pratiques de l'intégration continue

### Maintenir un gestionnaire de sources unique

Cela permet au code source de ne pas être éparpillé. Il permet également de garder un audit complet sur l’ensemble des changements qui ont eu lieu au fils des années.

Il existe plusieurs gestionnaires de source ; Git, SVN, CVS …

Le logiciel doit être construit à partir du gestionnaire de sources. Celui-ci doit être comme le lieu unique qui contient le code source de référence.

### Automatiser la construction (Build)

Le build d’un logiciel peut être un processus long et fastidieux. Son automatisation permet d’éviter les erreurs humaines (oublie de certains fichiers ou certaines étapes …) et de gagner un temps considérable que peut être alloué à des tâches à plus forte valeur ajoutée.

### Rendre le Build « auto-testant » pour capturer les bugs

Traditionnellement, un build signifie compiler, lier et toutes les autres tâches requises pour obtenir un programme exécutable.

Or un programme peut compiler mais cela ne signifie en aucun cas qu’il marche et qu’il est exempt de tout bug.

C’est pour cette raison qu’il faut associer à la phase de build une batterie de tests automatisés.

Ces tests sont notamment ce qu’on appelle les tests unitaires, des framworks tel junit et xunit aident à leur réalisation.

### Tout le monde soumet son travail (commit) au même endroit.