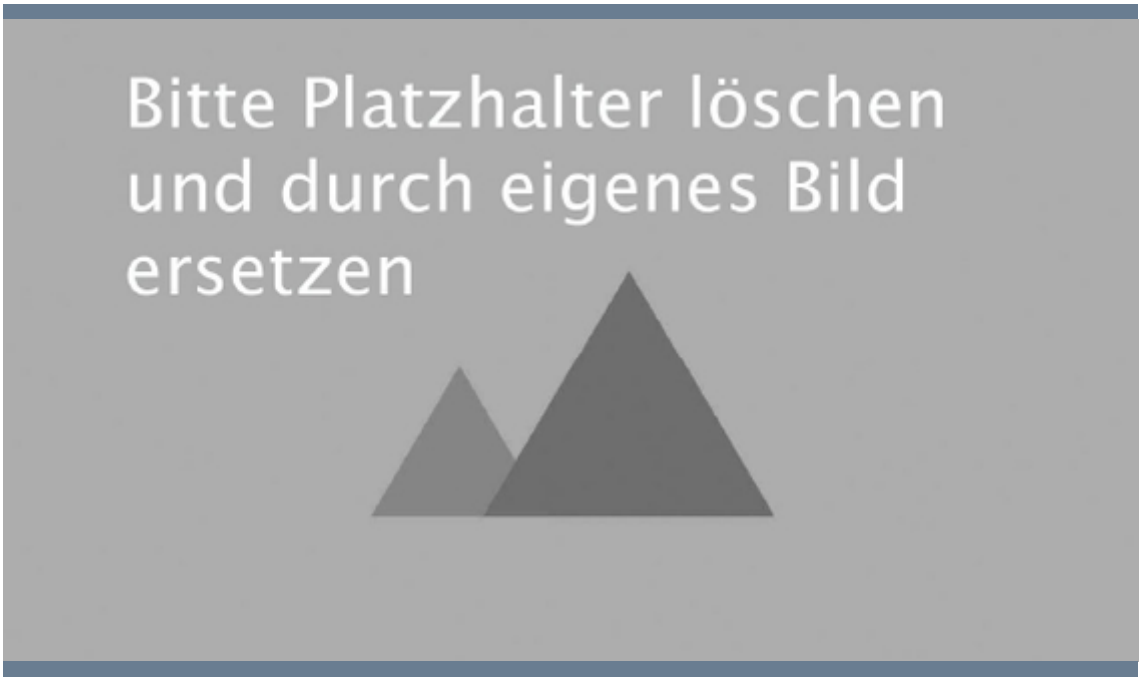


# Intégration continues

**Rapport 2 - Séminaire Info 2015**

Filière d'études :Informatique
Auteurs : Emanuel Knecht, David Aeschlimann
Conseiller : Dr. Bernhard Anrig
Date : 16 novembre 2015

# Abstract

# Table des matières

# 1 Introduction

Ce document est la partie écrite du module Séminaire Informatique de l'Haute école specialisée de Berne.

## 1.1 Mission

L'objectif de ce rapport est d'offrir un apercu de l'intégration continue (Continuous Integration) et des solutions existantes aux lecteur.

Dans une première partie la notion Intégration Continue et les concept correspondants seront expliquer. De plus il faut absolument mentionner les meuilleures pratiques de l'IC et les benefices qu'on recoit si on decide d'implémenter ces concepts.

Dans une deuxième partie du rapport on vous donneras une vue d'ensemble de tout les outils disponible pour pratiquer l'IC. à cause du nombre immense de different outils, il ne nous sera pas possible de considérer tous les composant existant. Le but est de démontrer les avantages et désavantages des quelques outils sélectionné, entre autres les outil les plus répandu.

## 1.2 Approche

Pour commencer la connaissance de la matière devait être acquisée et solidifiée. Dans notre parcours professionnel on a déjà encontrer des système de l'Intégration Continue, mais seulement comme utilisateurs et jamais comme administrateur. Après avoir definie la structure de notre rapport on a partagé les travaux et continué à travailler individuellement.

... à la fin (gegenlesen etc.)

Pour être capable à démontrer des differents serveurs de CI et mieux donner une évaluation, on a decidé de configurer et installer trois serveur en nuage. De plus on a créeé on projet de teste en java et c# pour illustrer un processus d'IC complet.

# 2 Intégration Continue

## 2.1 Aperçu

Le processus de developpement d'un logiciel. Ce que c'est CI, la definition ? Les concepts core, quels composant faut-il qu'on parle de CI ?

## 2.2 Histoire

## 2.3 Concepts

### 2.3.1 Construction continue

### 2.3.2 Intégration continue de base de donnée

### 2.3.3 Test continue

### 2.3.4 Inspection continue

### 2.3.5 Déploiment continue

-> Continuous Delivery concept

### 2.3.6 Information en retour continue

## 2.4 Motivation et bénéfices

La raison principale pour utiliser l'IC est de garantir le succès et le déroulement d'un projet de développement de logiciel sans accroc. Dans tous les projets il y auras des problème et dans tous les logiciels il y auras des bogues. Mais l'IC aide à minimiser l'impact negatif que cettes erreurs ont.

De plus il est possible d'automatiser des processus ennuyeux, répétitif et sensible aux défauts et comme ça économiser du temps et de la monnaie.

### 2.4.1 Éviter des risques

En dessous vous trouvez quelques risques que l'IC aide à éviter, mais seulement si elle est appliquée correctement (Meuilleures pratiques).

#### Logiciel pas déployable

Si on fait l'integration seulement à la fin du projét, la probabilité de ne pas être capable à déployer et dérouler le logiciel pour le client est très haute. Des énonces comme "Mais ça marche sur ma machine" sont très connues. Des raisons pour cela peuvent être des configurations manquantes ou differentes, ou même des dépendances qui n'ont pas été inclus. Naturellement si la source ne compile pas, le logiciel ne peut non plus être dérouler.

En commettre, construire et deployer le logiciel souvent ce risque peut être diminuer.

#### Découverte tarde des erreurs

#### Manque de visibilité du projet

#### Basse qualité de logiciel

## 2.5 Meuilleures pratiques

Peut-être c'est mieux de ne pas faire une intégration continue complet dans tous les cas. Pas introduire tout les concepts en meme temps, seulement si necessaire. Meuilleures pratiques

#### Commit code frequently

#### Dont commit broken code

#### Fix broken builds immediately

#### Write automated developer tests

#### All tests and inspections must pass

#### Run private builds

#### Avoid getting broken code

#### Decouple build process from IDE

# 3 Évaluation

## 3.1 Outils necessaire pour CI

## 3.2 Logiciels de construction

### 3.2.1 Ant

### 3.2.2 Maven

## 3.3 Serveur de l'intégration continue

### 3.3.1 Jenkins

### 3.3.2 TeamCity

### 3.3.3 Bamboo

### 3.3.4 Hudson

TFS, Cruise Control, Travis CI, Go

## 3.4 Autre outils

Test coverage, coding conventions, etc

# 4 Conclusion

## 4.1 Bilan

# Bibliographie

P. M. Duvall. *Continuous Integration : Improving Software Quality and Reducing Risk*, volume 1. Addison-Wesley Professional, 2007. ISBN 978-0321336385.

# Table des figures