

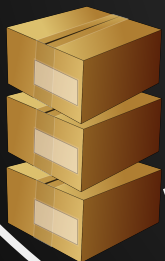
Continuous Integration

Produktentwicklungszyklus

Anforderungen

verstehen
bearbeiten
zerlegen

Features / Storys



Entwicklungsiteration (Sprint / Projekt)

programmieren
erstellen
verifizieren



releasen (veröffentlichen)
deploy (installieren)

Wo klemmt es?

Schmerzen:

- Fehlerhafte Software
- Grüner Build
- Integration

Negative Folgen in Bezug auf **Zeit** und damit auf **Kosten** und **Geschwindigkeit**

Ursache

Mainlinie veraltet durch mangelnde Integration
und Commits.

Ergebins?

- Auseinanderentwicklung des Codestandes
- Featurebranches nach Monaten zurück in den "Trunk" integriert
 - Integration nightmare
 - Merge-Konflikte -> raus-"mergen" von Funktionalität
- Code ist nach checkout nicht baubar/erstellbar
- Code läuft nur auf Entwicklermaschine nicht in Umgebung

Continuous Integration

Ziel:

- Entwickler entfernt sich NICHT zuweit von der "main line" des Codes
- es gibt immer eine stabile fehlerfreie Version der Software

Dadurch:

- weniger Zeitverschwendung bei der Suche nach Fehlern und deren Behebung
- keine Zeitverschwendung bei der Integration

Martin Fowler

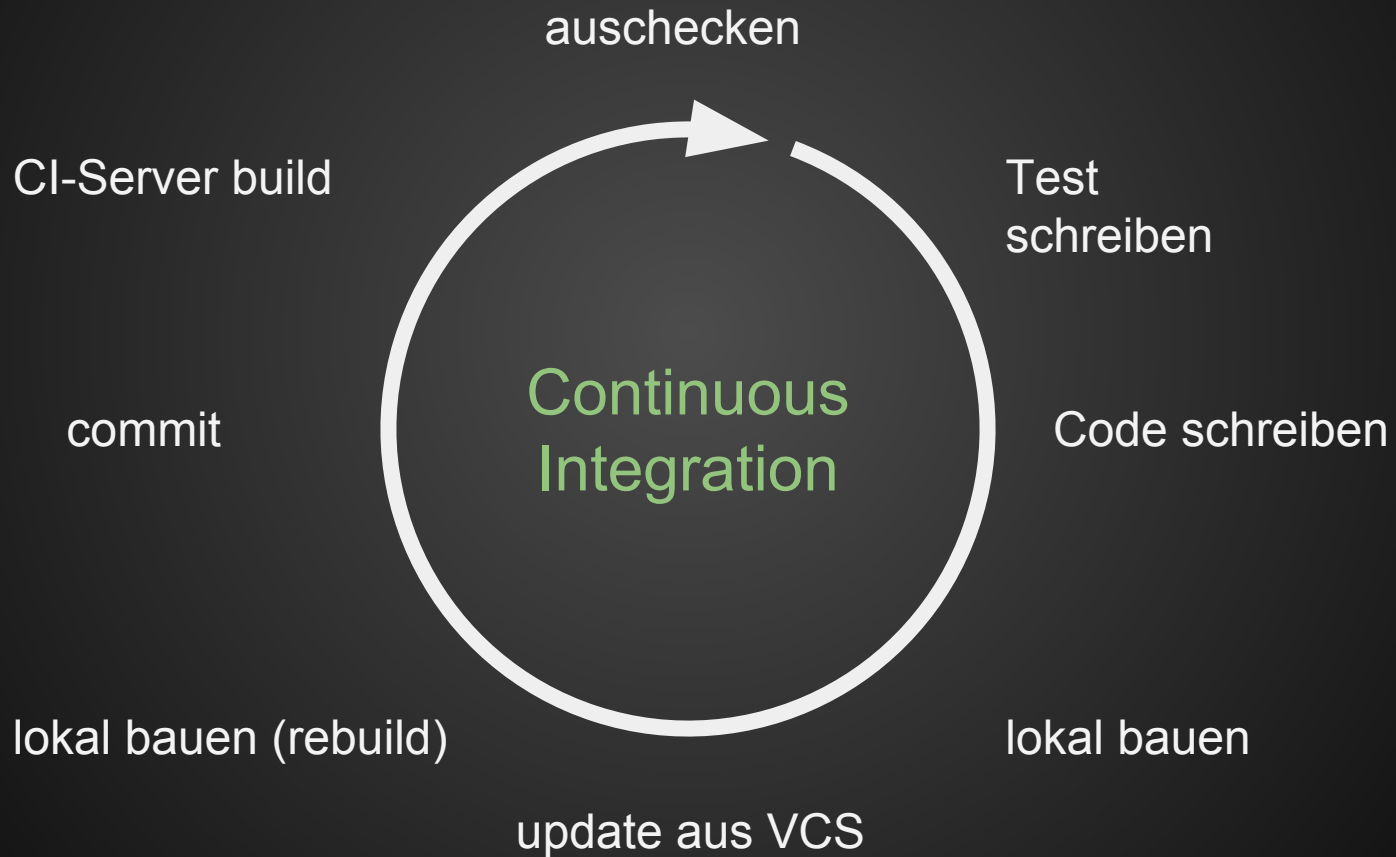
XP / CI / Software Architektur / Agiles Manifest

<http://martinfowler.com/articles/continuousIntegration.html>

Zyklus

- > auschecken -> Tests schreiben
- > Code schreiben -> lokal Build ausführen
- > update aus VCS -> rebuild -> einchecken
- > CI-Server führt Build aus

Zyklus im Bild



Ziel

Der Build bleibt immer grün (!), die Software ist immer lieferfähig (Bugfixes, etc.), keine Zeitverschwendung bei Integrationen.

Was gehört dazu

- VCS - verwaltet Hauptstrang (main line)
- Tests - automatisierte (!) Test /
Codeabdeckung > 90%
- Build Tools - lokale und automatisierte Build-Ausführung
- CI-Server - Continuous-Integration-Server

Die jungfräuliche Maschine

Checkout und Bauen des Projektes muss auf einer leeren Maschine funktionieren!

Ausnahme: Dinge die groß oder kompliziert zu installieren sind dürfen drauf sein z.B. OS, JDK, Datenbank

Build-Automation

- Erstellungsprozess muss kontinuierlich erfolgen.
- Angestoßen über Commit/Pull ins VCS
- Auschecken, Bauen, Tests ausführen, Datenbank aufsetzen, Test-Maschinen installieren, etc.

Lösung -> Continuous Integration Server

Continuous Integration Server

- Selfmade CIS - Cron-Job + Shell-Skript
- CruiseControl - der Veteran :-)

Frei

- Jenkins / Hudson

Kommerziell

- TeamCity
- Bamboo

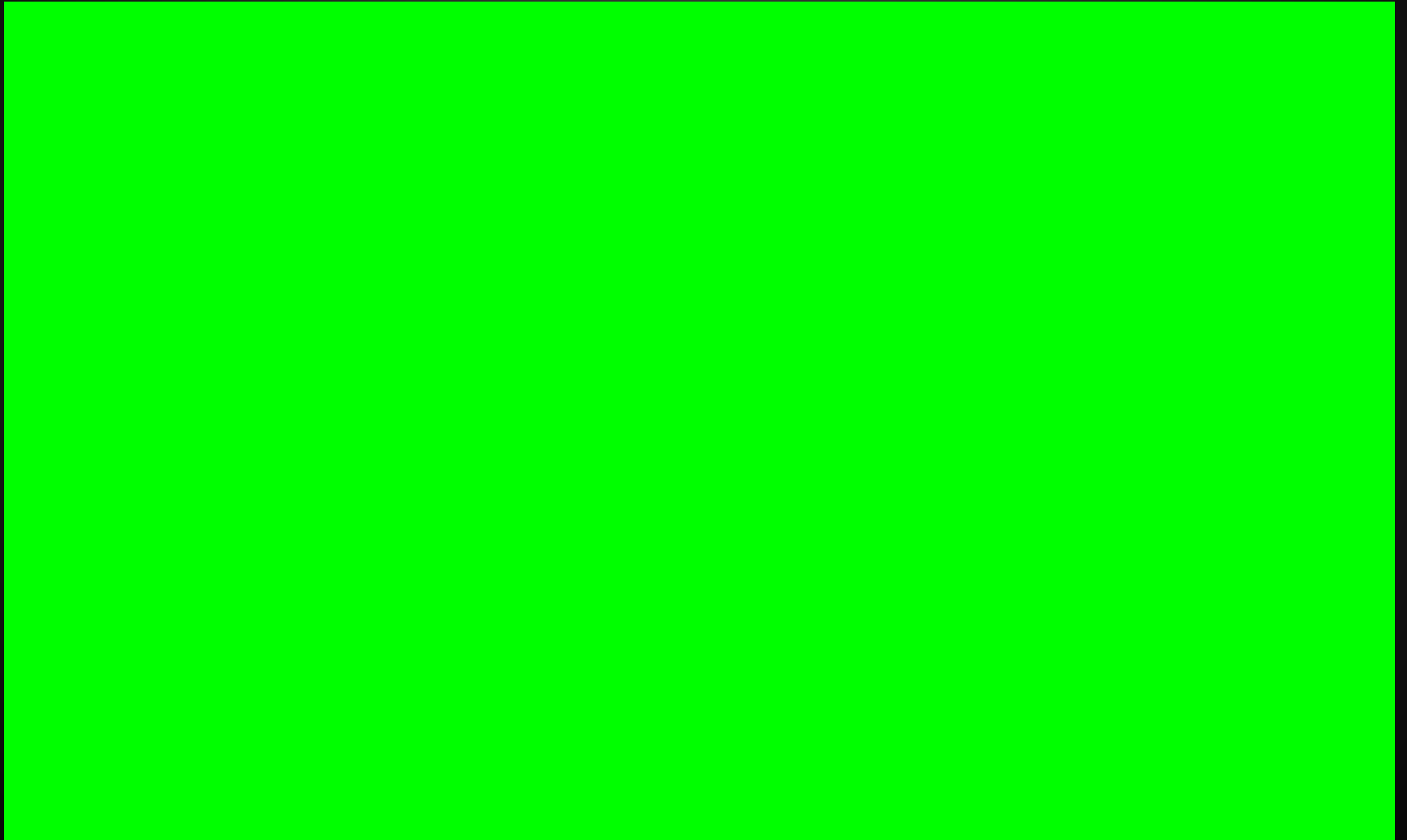
CI-Server in the cloud

- <https://travis-ci.org>
- <https://www.codeship.io>
- <https://drone.io/>

Regeln



- Der Build ist immer grün.
- Niemand commitet auf eine roten Build.
- Vor dem einchecken wird der Build lokal ausgeführt.
- Es gibt keinen Code ohne Tests.
(Ausnahme: Setter und Getter, etc.)
- Wenn der Build rot ist kümmern sich alle darum das er wieder grün wird.
- Niemand commitet ein abends und geht ohne zu prüfen ob der Build danach grün ist.

Kernaussage



Monitoring des Builds

Monitor im Team-Bereich der den Zustand des Builds anzeigt / visualisiert.

<div>✗ Jason Acceptance Test</div> <div>Z Personal Branches 3 months ago</div> <div>Updated by Jason Berry</div>	<div>✗ Mercurial Chain</div> <div>Z Personal Branches 1 month ago</div> <div>Manual build by Jens Schumacher</div>	<div>✓ Acceptance Test JDK 1.6</div> <div>Bamboo 12 hours ago</div> <div>Manual build by Przemyslaw Bruski</div>	<div>✓ CI Tests</div> <div> Bamboo 7 hours ago</div> <div>Manual build by Giancarlo Lionetti</div>
<div>✓ Deploy</div> <div>Bamboo 12 hours ago</div> <div>Scheduled build</div>	<div>✓ Remote Agent Functional</div> <div>Bamboo 4 hours ago</div> <div>Scheduled build</div>	<div>✓ Stable CI Tests</div> <div> Bamboo 43 minutes ago</div> <div>Updated by Brydie McCoy</div>	<div>✓ Atlassian Command Line</div> <div>Bamboo Plugins 7 hours ago</div> <div>Dependant of TRIGGER-TRUNK-2061</div>
<div>✓ Bamboo FindBugs Plugin</div> <div>Bamboo Plugins 7 hours ago</div> <div>Dependant of TRIGGER-TRUNK-2061</div>	<div>✓ Bamboo JMeter Aggregate</div> <div>Bamboo Plugins 6 hours ago</div> <div>Dependant of TRIGGER-TRUNK-2061</div>	<div>✓ Bamboo Sandbox Plugin</div> <div>Bamboo Plugins 6 hours ago</div> <div>Dependant of TRIGGER-TRUNK-2061</div>	<div>✓ Bamboo Sonar plugin</div> <div>Bamboo Plugins 6 hours ago</div> <div>Dependant of TRIGGER-TRUNK-2061</div>

Weitere Themen

- Nightly Builds - keine gute Idee -> Bugs bleiben lange unentdeckt
- Geschwindigkeit des Build - je schneller desto besser -> Schnelles Feedback
 - < 5 min. nach Fowler < 10min.
- Staged Build / Deployment Pipeline
 - Aufteilung des Build in Schritte
 - Schnelles Feedback langsame Build-Schritte später
- Tests
- automatisiertes Deployment

Staged build / deployment pipeline

The screenshot displays the Jenkins web interface for a project named "Registration API (PAR) (Team Joker)". The top navigation bar includes links for "Projects", "My Changes", "Agents (13)", and "Build Queue (1)". The main content area shows a list of build stages, each with a dropdown arrow and a "Run" button. The stages are:

- 1. Build + Test + RPM + DEV-Repo (Run button)
- 1b. Code Quality (Run button)
- 2. Deploy DEV (Run button)
- 3. ITs on DEV + RC-Repo (Run button)
- 4. Deploy RC (Run button)
- 5. ITs on RC + CR-Repo (Run button)

Each stage is represented by a row with the following details:

Build Number	Status	Tests	Artifacts	Agent	Time
#539	Success	Tests passed: 205, ignored: 1	No artifacts	lgentsch (1)	13 hours ago (1m:54s)
#411	Success	Tests passed: 205, ignored: 1	No artifacts	lgentsch (1)	13 hours ago (4m:40s)
#555	Success	Success	No artifacts	lgentsch (1)	13 hours ago (2m:29s)
#730	Success	Success	No artifacts	lgentsch (1)	13 hours ago (6m:42s)
#671	Success	Success	No artifacts	lgentsch (1)	13 hours ago (2m:01s)
#743	Success	Success	No artifacts	lgentsch (1)	12 hours ago (6m:28s)

Literatur

Links:

<http://www.jamesshore.com/Blog/Continuous-Integration-is-an-Attitude.html>

<http://martinfowler.com/articles/continuousIntegration.html>

Bücher:

Continuous Integration: Improving Software Quality and Reducing Risk

Fragen?