



Auswahl eines Continuous Integrationservers



Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de

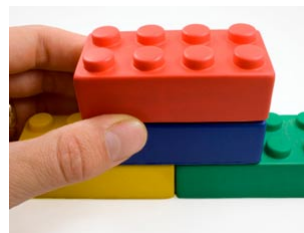
Version: 1.0

Gliederung

- Einführung
- Auswahlkriterien
- Fazit

- **Einführung**
- Auswahlkriterien
- Fazit

- Anwendung regelmäßig vollständig neu Bauen und Testen
- Populär gemacht durch Kent Beck, Martin Fowler
- Vorstufe: Nightly Build
- Ziel: Qualität steigern
- OIO setzt seit ~2001 CI ein
- OIO sucht aktuell neuen CI-Server



Was leistet ein CI-Server?



Bauen



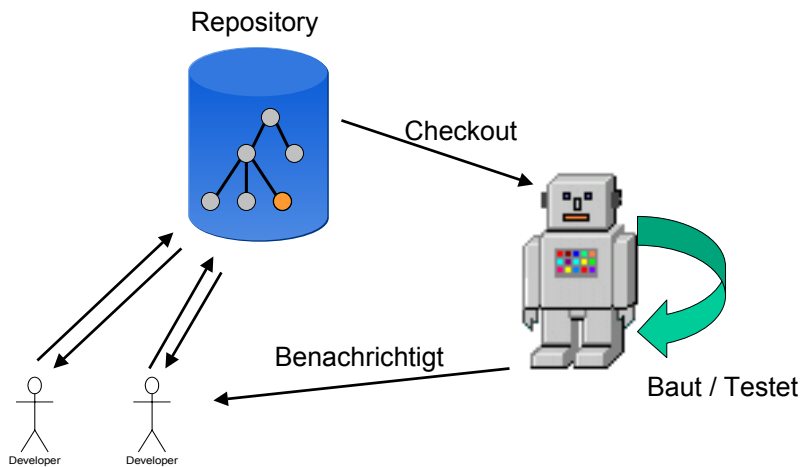
Testen



Reporten

5

Arbeitsweise eines CI-Servers



6

Vorteile von Continuous Integration

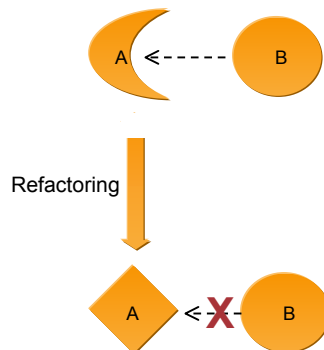
- Deckt Probleme bei Integration einzelner Bestandteile auf
- Probleme werden zeitnah (kontinuierlich) entdeckt
- Kein "Vergessen zu Testen" mehr
- Stets lauffähige Version
- Erzieherische Wirkung auf Entwickler



7

Integrationsproblematik

- Tests einzelner Module OK
- Gesamtsystem nicht lauffähig



8

Gliederung

- Einführung
- **Auswahlkriterien**
- Fazit

9

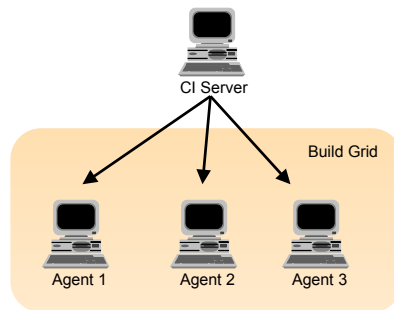
Source Code Management

- **Mainstream**
 - CVS
 - SVN
- **Distributed SCM Tools**
 - Mercurial
 - GIT
- **passende SCM Unterstützung ist KO-Kriterium**



10

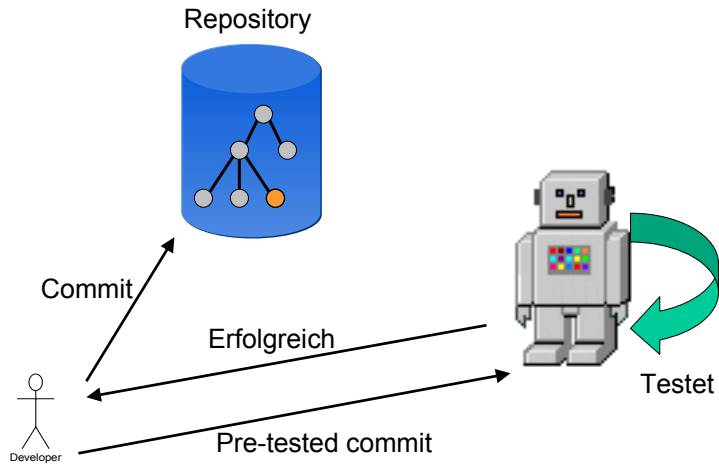
- Distributed Builds
 - Bauen und testen in verschiedenen Umgebungen
 - **Verschiedene: JDK, Betriebssysteme, Application Server, Browser, Datenbanken.**
 - Schnellere Builds durch Lastverteilung
- Parallel Builds



11

- Triggered Builds
 - Manuell
 - SCM gesteuert
 - Zeitplan gesteuert
- Multiproject Builds
 - Manuelle Abhängigkeiten
 - Automatisches Erkennen und Bauen von Abhängigkeiten
 - **Maven**
 - **Ivy**
- Reproduzierbare Builds

12

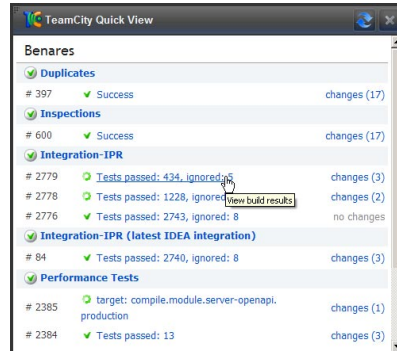


13

- Offener CI Server
- Sicherheitssystem
 - Benutzer und Gruppenverwaltung
 - Projektabhängige Rechteverwaltung für Benutzer und Gruppen
 - Anonyme Zugänge
- Externe Authentifizierung
 - LDAP

14

- Benachrichtigung über erfolgreiche / fehlgeschlagene Builds
- Email
- RSS
- Jabber, ICQ, ...
- Windows System Tray
- IDE



15

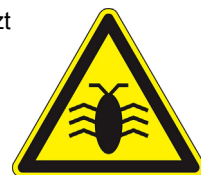
- Leicht zu bedienen
- Voller Zugriff auf die Konfigurationsmöglichkeiten
- Übersicht aller Projekte auf einen Blick
 - Status
 - Manuelles starten und stoppen des Builds
- Ansichten für
 - Reports
 - Quellcodedarstellung
 - Konsolenausgabe für jeden Build

16

- Native Unterstützung für die populärsten Build Tools
 - Ant, Maven, GANT, Rake, Nant, Shell
- Möglichkeiten weitere Build Tools zu installieren
- Verwenden von mehreren Build Tools gleichzeitig
 - Kombination aus Maven und Ant

17

- Verknüpfen von Issue Tracker und CI Server
- Im CI-Server
 - Erfolgreicher Build
 - **Welche neuen Features/Fixes sind in einem Build enthalten**
 - Fehlgeschlagener Build
 - **Wer/Was ist verantwortlich**
- Im Issue Tracker
 - In welchem Build wurde das Issue erfolgreich umgesetzt
- Verknüpfung
 - durch Entwickler -> Commit Message
 - Mylyn generierte Commit-Messages




18

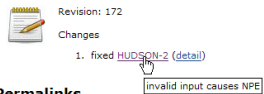
Issue/Bug Tracking (2)

- Auswahlkriterien
 - Tools
 - **Bugzilla**
 - **Trac**
 - **JIRA**
 - Uni-/Bidirektional
 - **CI Server updated Issue automatisch**
 - **Issue linkt auf CI Build**

19

Issue/Bug Tracking (3)

- JIRA Plugin für Hudson
 - In Hudson:
 -  **Build #14 (Mar 10, 2007 5:41:40 PM)**

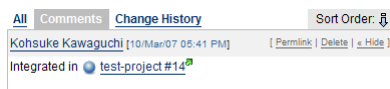


Revision: 172
Changes
1. fixed HUDSON-2 (detail)

Permalinks

- [Build number](#)

- In JIRA:



All | Comments | **Change History** | Sort Order: ⌵

Kohsuke Kawaguchi [10/Mar/07 05:41 PM] [Permalink] [Delete] [Hide]

Integrated in [test-project #14](#)

20

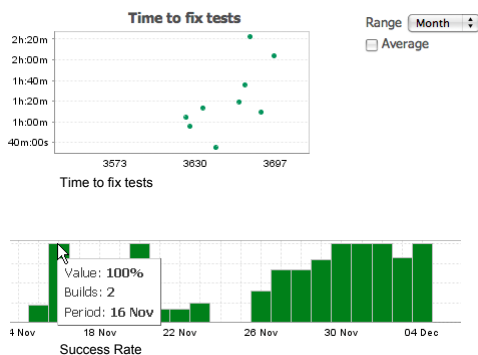
Test Tool Integration

- Result rendering
 - JUnit
 - Selenium / Webtest
 - PMD, Checkstyle
 - Code Coverage
- Auswirkung auf Build-Status
- Verknüpfung mit IDE

21

Reporting

- Reports von "Techi-Tools"
- Statistiken, Metriken
 - Macht CI Server zu Management-Tool



22

IDE Integration



- Builds überwachen / Status einsehen
- Build-Log einsehen
- Builds anstoßen
- Übersicht über fehlgeschlagene Tests
 - Tests direkt anspringen
- Verknüpfung mit WebInterface
- (Personal Builds, Pre-tested Commit)

23

Plugin Schnittstelle

- Große Auswahl an fertigen Plugins
- Einfache Installation und Update
- Umfangreiche und gut dokumentierte API

24

Community

- Aktivität, Entwicklungsgeschwindigkeit
- Mailingliste, Forum, Wiki
- Direkter Kontakt zu den Entwicklern
- Issue Tracker

25

Gliederung

- Einführung
- Auswahlkriterien
- **Fazit**

26

TeamCity vs. Hudson

- Teamcity
 - Closed-Source
 - Lizenzgebühren
 - Statistiken
 - Pre-Tested Commit
 - Sehr gute IDE-Integration
 - Plugin-Schnittstelle
 - wenige Plugins vorhanden
 - Keine JIRA-Integration
- Hudson
 - Open-Source
 - kostenlos
 - Keine Statistiken
 - Kein Pre-Tested Commit
 - Schlechte IDE-Integration
 - Plugin-Schnittstelle
 - viele Plugins vorhanden
 - JIRA Integration

27

Fazit

- Neue CI-Generation
 - Server bieten viele neue Features
 - Einstiegshürde und Komplexität sinkt
- Es gibt nicht DEN besten CI Server
- Eigene Schwerpunkte finden -> passenden CI Server wählen
- Feature-Matrix für groben Überblick
- Favoriten evaluieren

28

- CI Feature Matrix von ThoughtWorks
 - <http://confluence.public.thoughtworks.org/display/CC/CI+Feature+Matrix>
- Hudson Website
 - <https://hudson.dev.java.net/>
- Artikel von Martin Fowler
 - <http://www.martinfowler.com/articles/continuousIntegration.html>
- OIO-Artikel: CI mit CruiseControl
 - <http://www.oio.de/cruisecontrol.htm>



Vielen Dank für Ihre Aufmerksamkeit !

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de



Fragen ?

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de

Version: 1.0