Martin Heider, infomar software

Im Fluss bleiben

Continuous Integration mit Hudson

Agenda

Kontinuierliche Integration

- Ziel
- Grundprinzipien
- Adressierte Risiken
- Besser, schneller, günstiger
- Der Build und seine Inhalte
- Wie starten?
- Mit 7 Schritten in den CI Himmel

Agenda

Hudson

- Historie …
- Funktionsweise
- Einrichtung
- Features & Plugins
- Gemachte Erfahrungen

Ein kleiner Film zum Schluss

Kontinuierliche Integration
Ziel

Wir wollen
Software … entwicklen

Günstiger

Transparenter

Schneller

Zuverlässiger

Besser

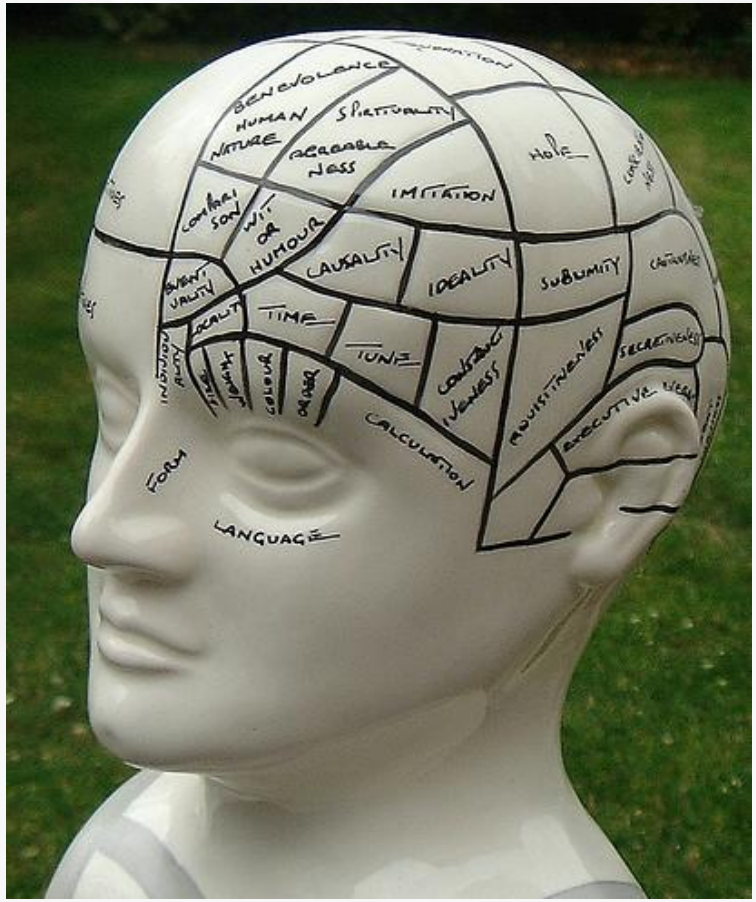
Kontinuierliche Integration
Grundprinzipien

Integration
ist aufwändig
darum machen wir
es jetzt öfter ☺

Kontinuierliche Integration
Grundprinzipien

Entwicklungsmethode

- Mehr Mindset als Technik
- Unterstützt von Tools



Kontinuierliche Integration
Grundprinzipien

Entwickler integrieren fortlaufend

- Kein Problemstau
- Im Fluss bleiben

Kontinuierliche Integration
Grundprinzipien

Vollautomatisierter Build

- „One green button“
- „Stop the line“

Kontinuierliche Integration
Grundprinzipien

Was es nicht ist

- Nightly builds
- Entwickler branches
- Vereinbarte Integrationstermine
- Bauen mit der IDE

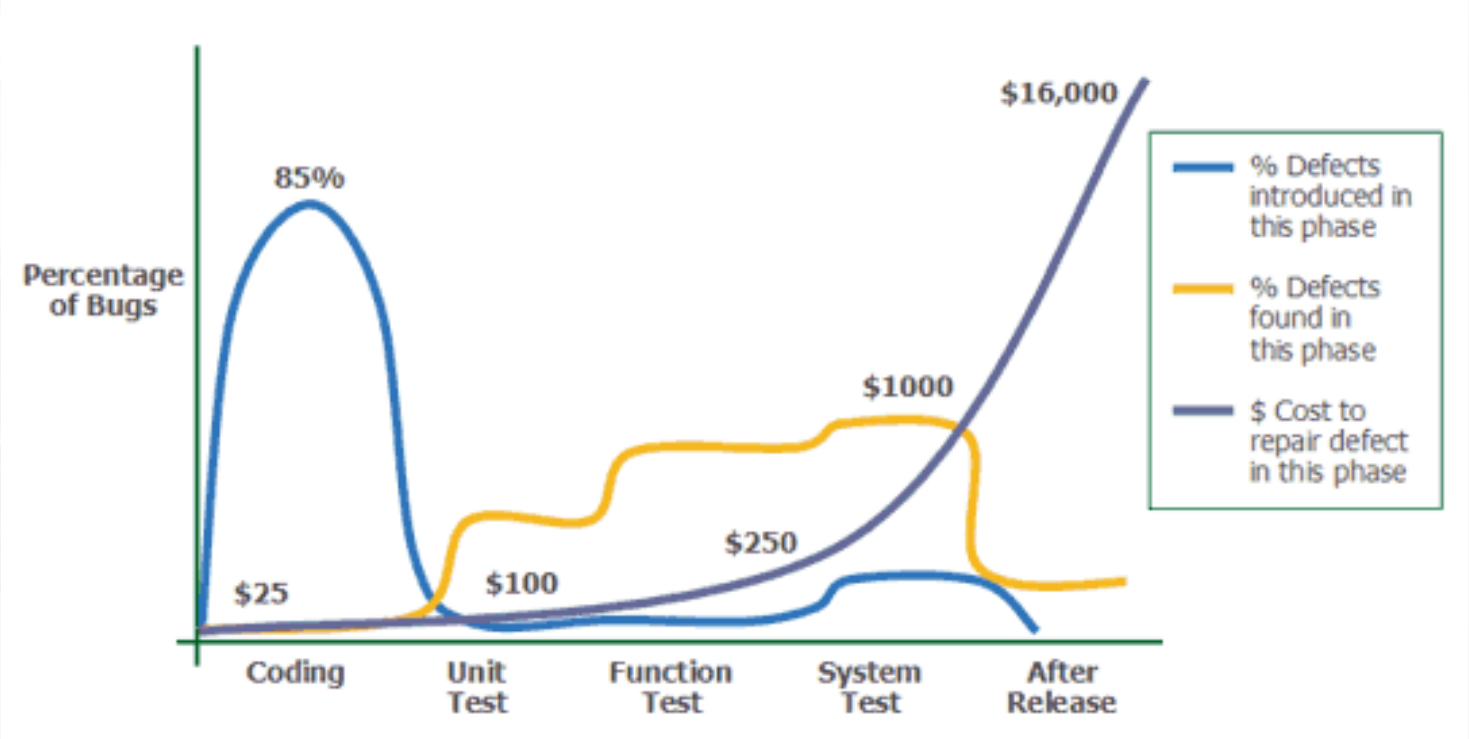
Kontinuierliche Integration
Addressierte Risiken

zur bisherigen Integration

Zu Risiken fragen Sie Ihren
Projektleiter oder Integrator

Kontinuierliche Integration
Addressierte Risiken

Risiko I: Späte Fehlerbehebung ist teurer



http://www.agitar.com/solutions/why_unit_testing.html

Kontinuierliche Integration
Addressierte Risiken

Risiko II: Mangelnde Teamabstimmung

- „Deine Änderung passen nicht mit meinen zusammen“
- „Hattest Du das nicht bereits vor 2 Monaten gefixt“

Risiko III: Schlechte Code-Qualität

- „Wieso machen drei verschiedene Klassen das Gleiche?“
- „Der Code von Team XY schaut ja ganz anders aus“

Kontinuierliche Integration
Addressierte Risiken

Risiko IV: Mangelnde Transparenz / Sichtbarkeit

- „Welche Tests laufen nicht?“
- „Was beinhaltet Build 1.2.3?“
- „Wo stehen wir mit der Code-Abdeckung?“

Risiko V: Nicht verfügbare Software

- „Bei mir geht‘s“
- „Eigentlich läuft‘s“
- „Ich brauche noch einen Build zum Testen“
- „Morgen kommt der Chef-Chef, wir brauchen eine Demo“

Kontinuierliche Integration

Besser, schneller, günstiger

Kontinuierliche Integration
Besser, schneller, günstiger

Besser

- Oft und frühzeitig getestet
- „Coding Standards“ und „Best Practises“ einhaltend

Schneller

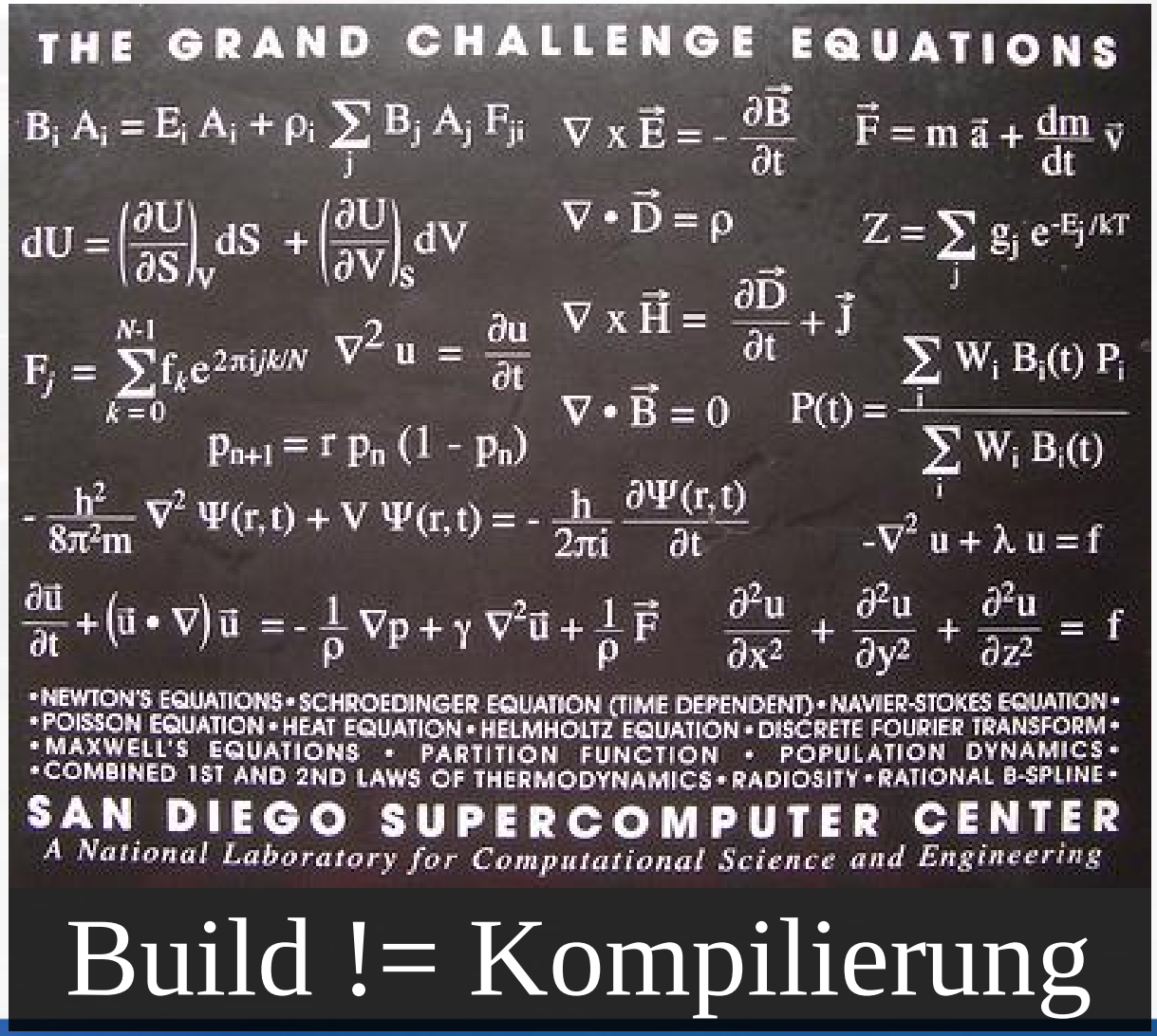
- Tests finden parallel zur Entwicklung statt
- Aufwändige Integrationen werden zum „Nicht-Ereignis“

Günstiger

- Fehler werden früher gefunden
- Behebung der Fehler zum frühesten, günstigsten Zeitpunkt
- Einfach wiederholbare Tests

Kontinuierliche Integration

Der Build und seine Inhalte …



Kontinuierliche Integration

Der Build und seine Inhalte ...

- Kompilierung
- Test Ausführung (Unit Tests, Akzeptanztests, etc.)
- Integration (Datenbank, Drittsysteme)
- Statische Analysen (Code & Architektur)
- Automatisches Deployment auf Staging / Produktiv Server
- Generierung der Dokumentation

Kontinuierliche Integration
Wie starten?

Kontinuierliche Integration
Wie starten?

Wann baue ich?

- Nach jeder Code Änderung
- Nach jeder Änderung von Abhängigkeiten

Wie baue ich?

- Mit einem einzigen Build-Skript
- Startbar auf der Kommandozeile
- Nicht in Abhängigkeit einer IDE

Was brauche ich dazu?

- Konfigurationsmanagement Software
- CI Software und Server

Kontinuierliche Integration
Wie starten?

Worauf ist noch zu achten?

- Schnelles Feedback
- Einfach zugreifbar
- Kein Aufwand für Entwickler
- Schlüsselmetriken identifizieren
- Wichtiges deutlich visualisieren
- Auf Schlüsselmetriken sofort reagieren

Kontinuierliche Integration
7 Schritte

7. Artefakte für **jeden** Build erstellen

6. Auf **allen** Zielplattformen bauen

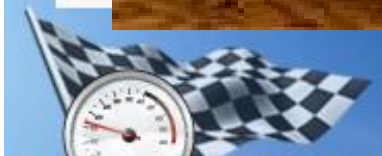
5. Aufgrund von Metriken **(re)agieren**

4. Probleme **früh angehen** und **schnell scheitern**

3. Build Fehler **sofort** beheben

2. Keinen Code einchecken, der nicht läuft

1. Früh und **oft** einchecken



Hudson


2008 DUKE'S CHOICE AWARDS
Java
WINNER


SOA WORLD
2008
SOA
READERS'
CHOICE
AWARD
RCA FINALIST
SYS-CON

Hudson
Historie ...

Vorfahren & Verwandte

- Urvater CruiseControl
- Vater Kohsuke Kawaguchi
- Continuum, Bamboo, TeamCity, Luntbuild, AnthillPro

Geburtsort und Datum

- Sun Microsystems, seit 2005, aktuell 1.300

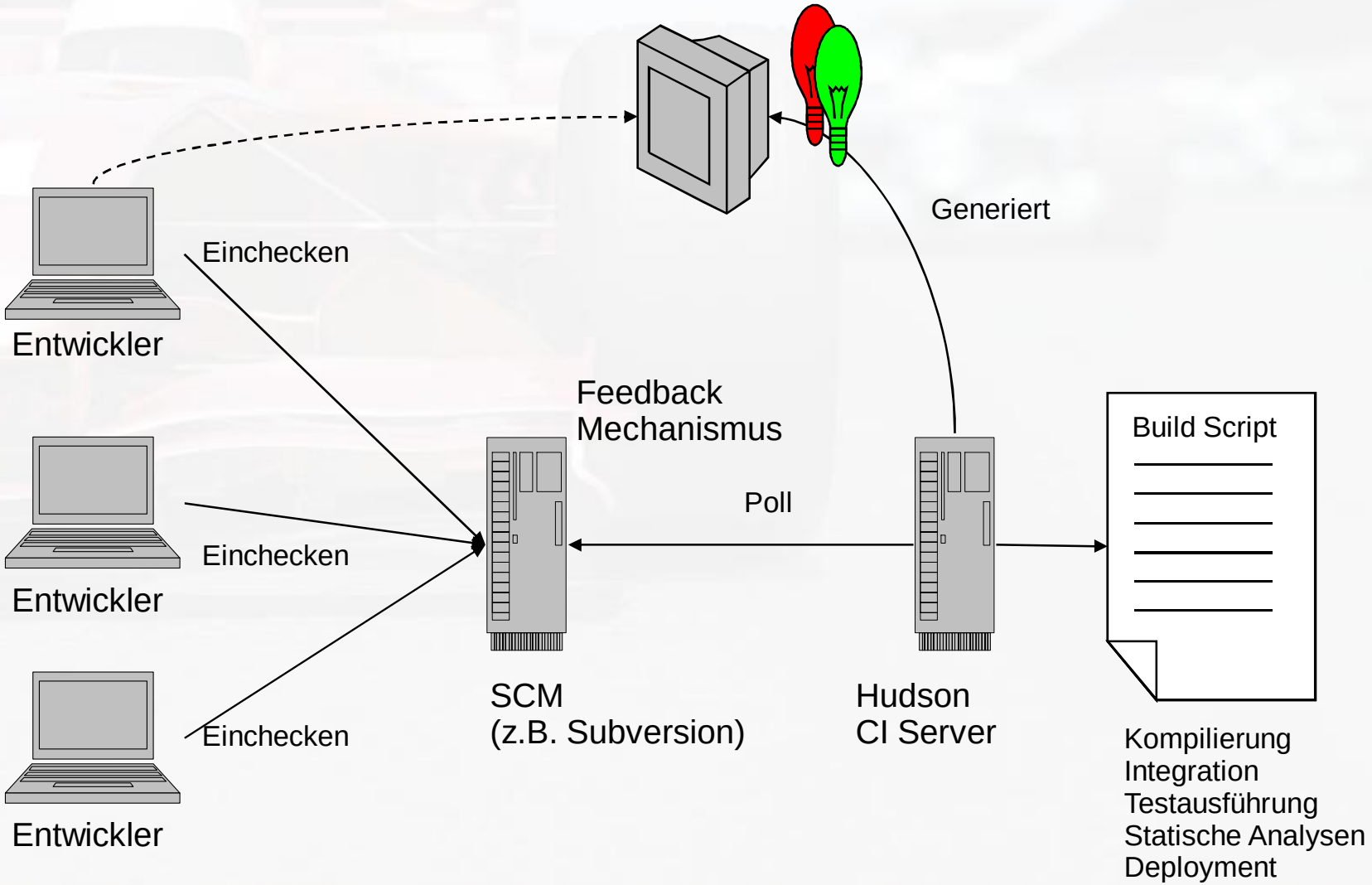
Geburtsurkunde (Lizenzmodell)

- Mischung aus MIT und Commons Creative

Notwendig für die ersten Schritte

- J2SE 1.5

Hudson
Funktionsprinzip



Hudson
Funktionsprinzip

- Sehr schnell “Up und running” (~ 1 Stunde)
  - Konfiguration des Hudson Servers
  - Einrichtung erster Build Jobs
- Skalierbar
  - Master & Slave
  - Einbindung Virtuelle Maschinen
  - Pipe Lining von Jobs
- Erweiterbar
  - Umfangreicher Plugin Support (135)

Hudson
Live Demo

http://localhost:8080

Hudson
Einrichtung allgemein

Hudson installieren & verwalten

- System konfigurieren
- Plugins & Knoten verwalten
- Als Service installieren

Betrieb

- Neu laden
- Herunterfahren vorbereiten

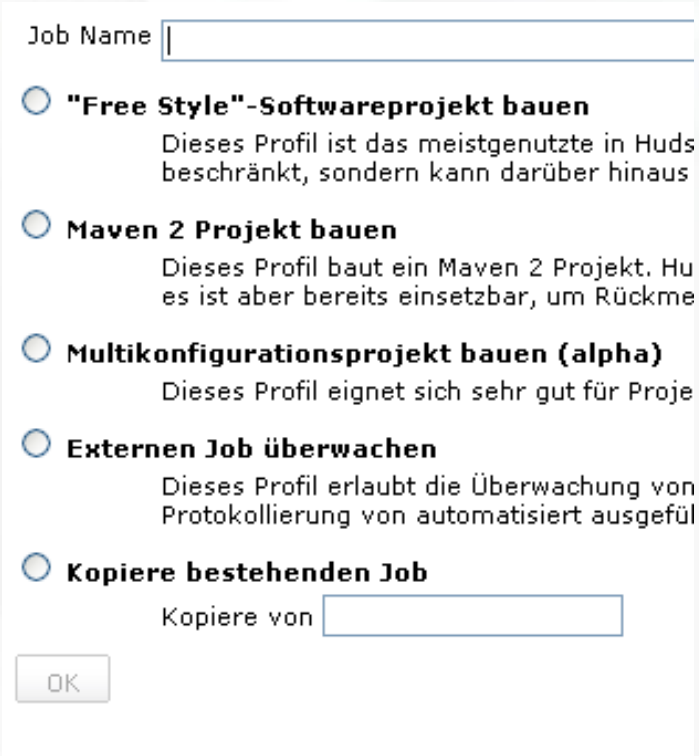
Diagnose

- Logging, System Log, Statistik
- Skript-Konsole

Hudson
Einrichtung Jobs

Job Typen

- Free Style
- Maven 2
- Multikonfigurationsprojekt
- Externen Job überwachen



Hudson
Einrichtung Jobs

Multikonfigurationsprojekt

- Definition von Achsen
- Varianten der Achsen werden als Variablen an den Build übergeben
- Alle Varianten werden durch mehrfachen Aufruf des gleichen Build-Skripts abgearbeitet

Hudson
Einrichtung Jobs

Externe Job überwachen

- Besteht lediglich aus Namen
- Benachrichtigung über Job Ausführungen erfolgt mittels
  - XML:

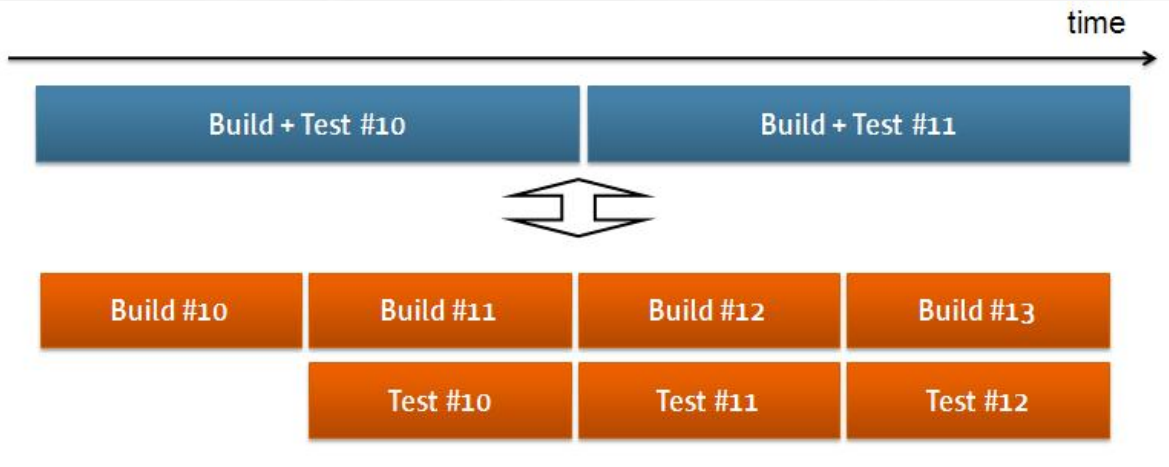
```
<run>
  <log encoding='hexBinary'>...</log>
  <result>integer. 0 is success and everything else is failure</result>
  <duration>milliseconds it took to execute this run</duration>
</run>
```

  - HTTP: http://myhost/hudson/job/<_jobName_>/postBuildResult.

Hudson
Einrichtung Jobs

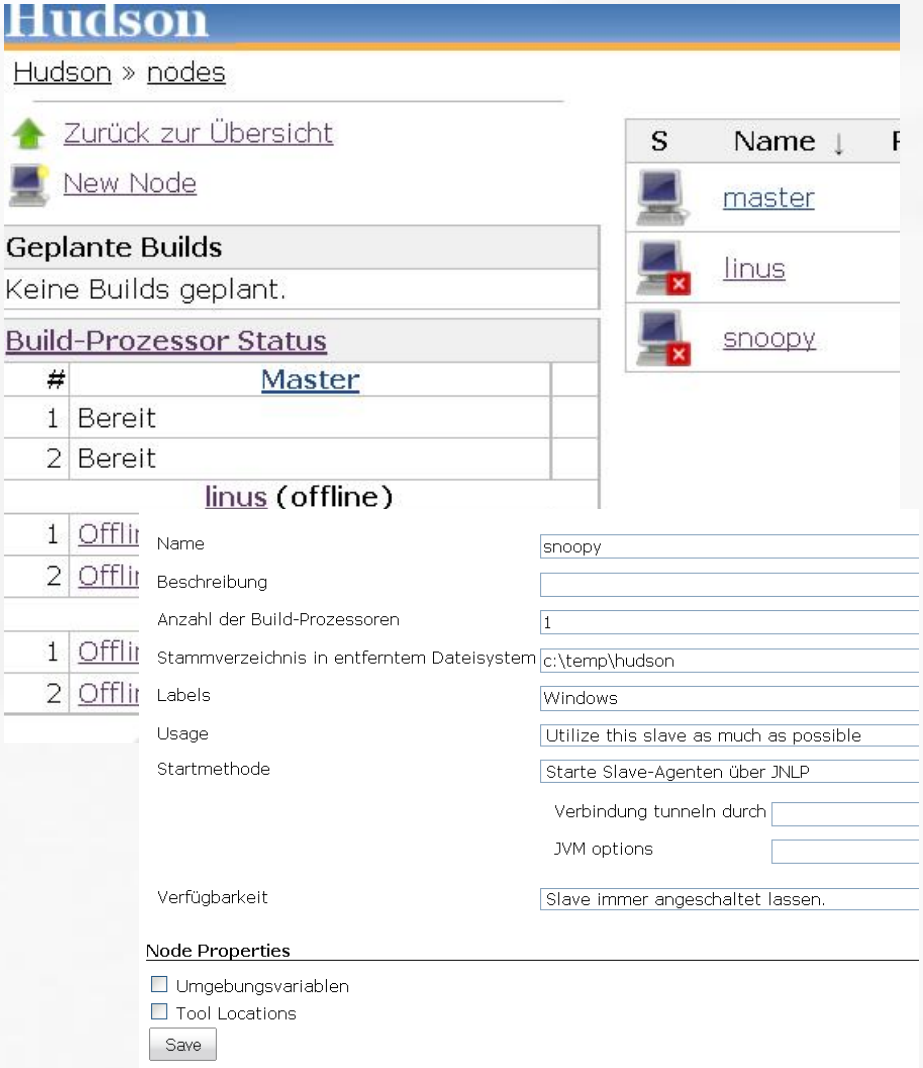
Jobs miteinander verknüpfen (Pipe Lining)

- Vor- oder nachgelagerte Projekte definieren
- Artefakte in vorgelagerten Projekten archivieren
- Archivierte Artefakte in nachgelagerten Projekten holen
- Fingerprinting von Artefakten in vor- und nachgelagerten Projekten hilft bei deren Zuordnung



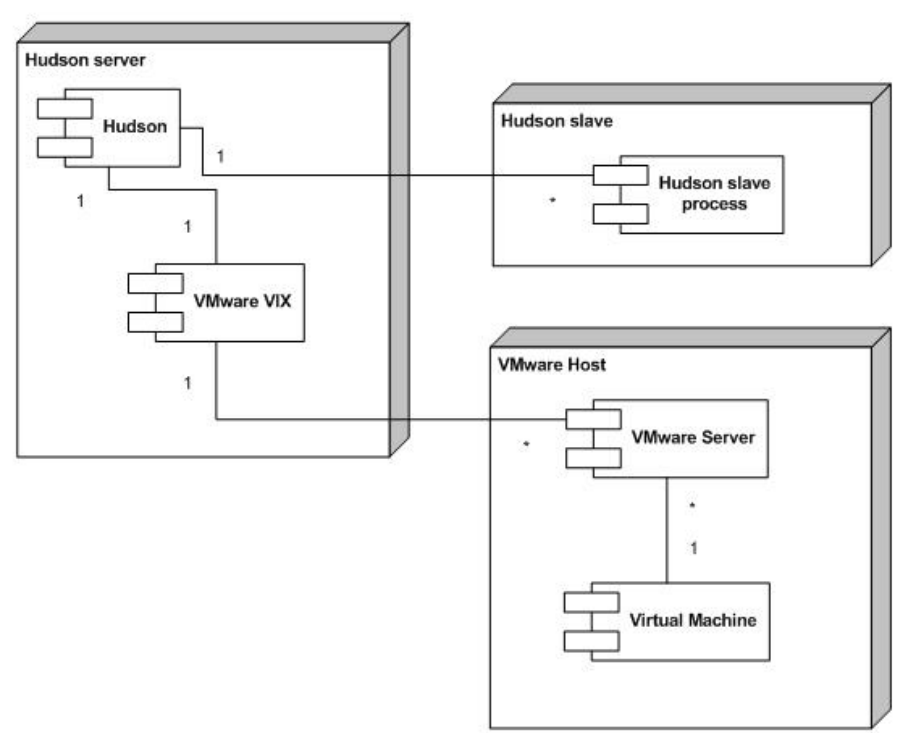
Hudson
Einrichtung Master & Slave

- Konfiguration der Slaves auf Master
- Einrichtung der Slaves
- Abstimmung Jobs & Slaves, damit Builds auf korrekte Maschinen kommen
- Master verteilt Builds



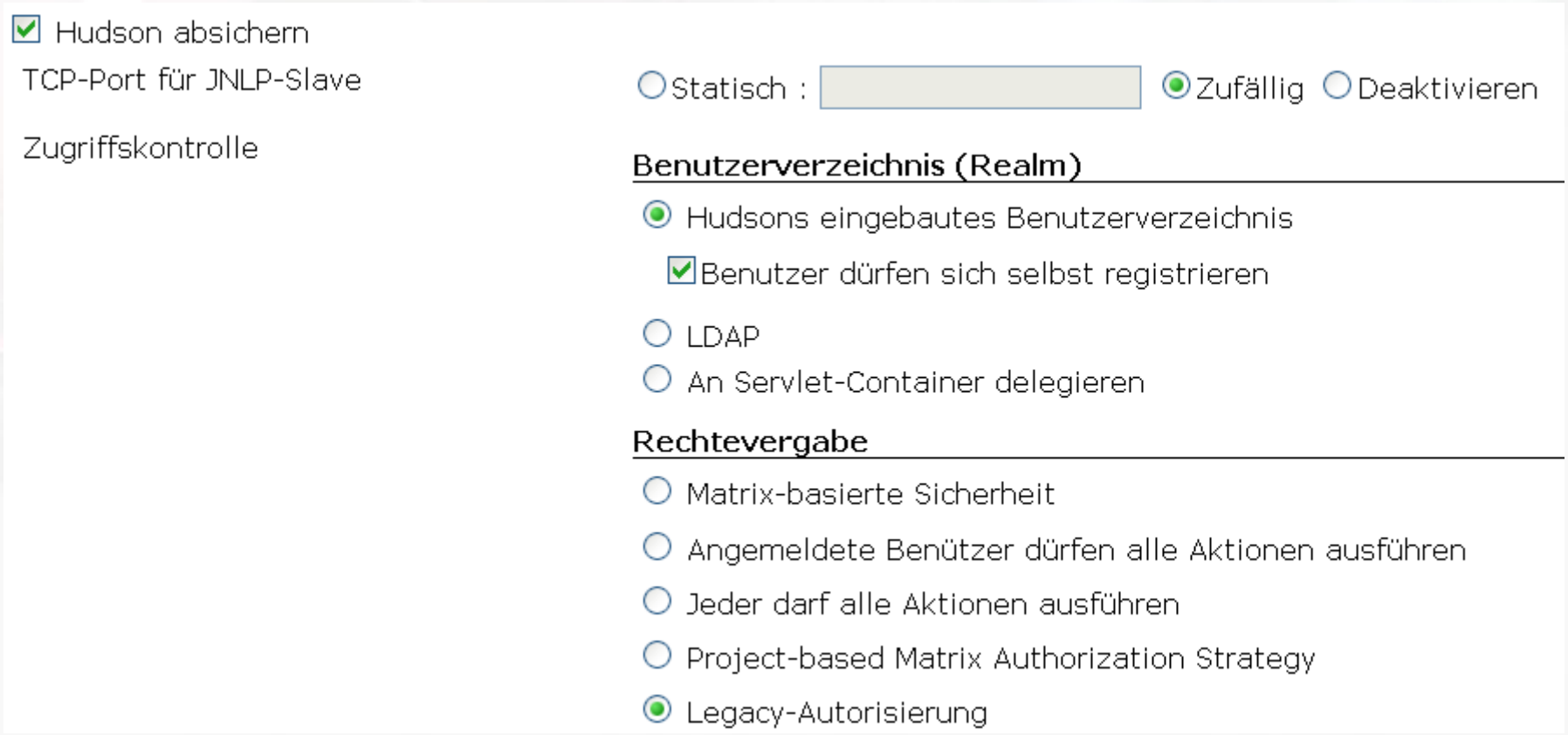
Hudson
Einrichtung virtuelle Welten

- Konfiguration des Hudson Servers zur Kommunikation mit Vmware Host
- Konfiguration der Jobs zur Zusammenarbeit mit bestimmten virtuellen Rechnern
- Aktuell auf einen Host begrenzt (besser wäre Schnittstelle zu VMware Virtual Infrastructure Server)



Hudson
Einrichtung Zugriff

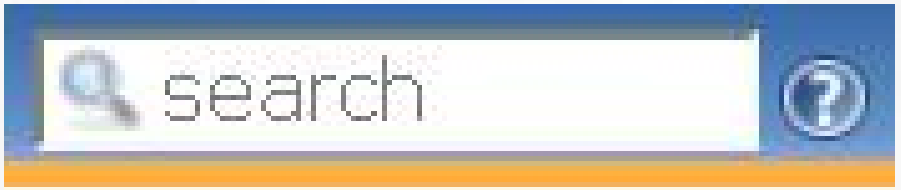
Benutzer & Rechte



Hudson
Features & Plugins

Schnelle Navigation über …

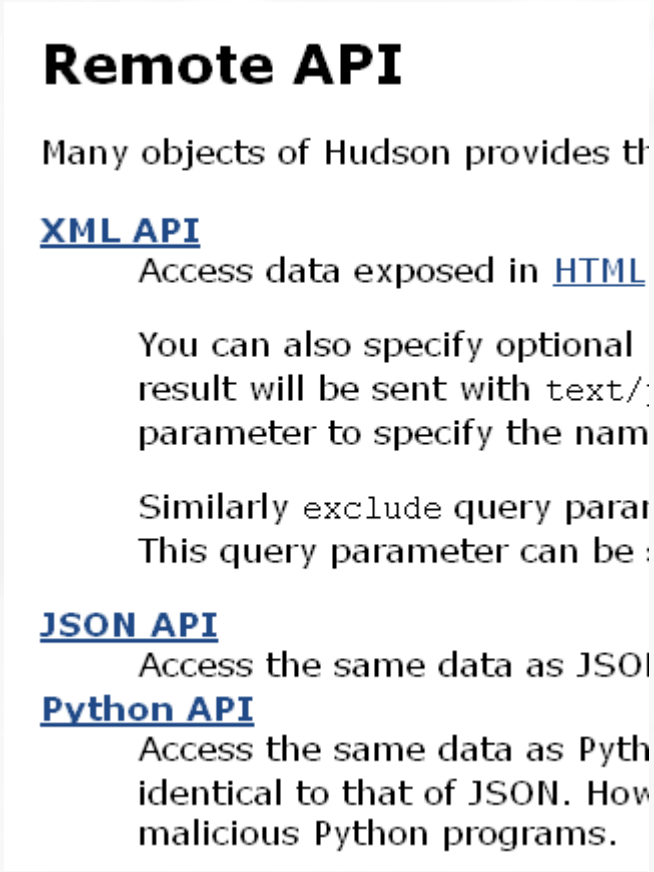
- Knotentypen, z.B. job, ws, changes, configure, etc.
- Über Knotennamen, z.B.
  - Job Name
  - Build Nummer
  - Slave Rechner



Hudson
Features & Plugins

Remote Access API

- REST Style
- Ermittlung von Daten für programmatische Weiterverarbeitung
- Anstoßen eines Builds
- Erzeugung oder Kopieren von Jobs



Hudson
Features & Plugins

- SCMs (GIT, Mercurial, ClearCase, etc.)
- Benachrichtigungsmechanismen
  (z.B. RSS, E-mail, IM, Twitter, GoogleCalendar, etc.)
- Statische Analysen (z.B. PMD, Checkstyle, Sonar, etc.)
- Tests (z.B. JUnit, TestNG, Selenium, JMeter, Grinder, etc.)
- Integration mit anderen Sites/Tools (z.B. Jira, Mantis, etc.)
- Meine Lieblinge
  - Permanente Links (z.B. für LATEST builds)
  - Nachträgliches Tagging von Builds
  - Trends
- Spezielles (z.B. Claim, Task Scanner, CI Game)

Hudson
Gemachte Erfahrungen

Wo ist noch Potential ☹

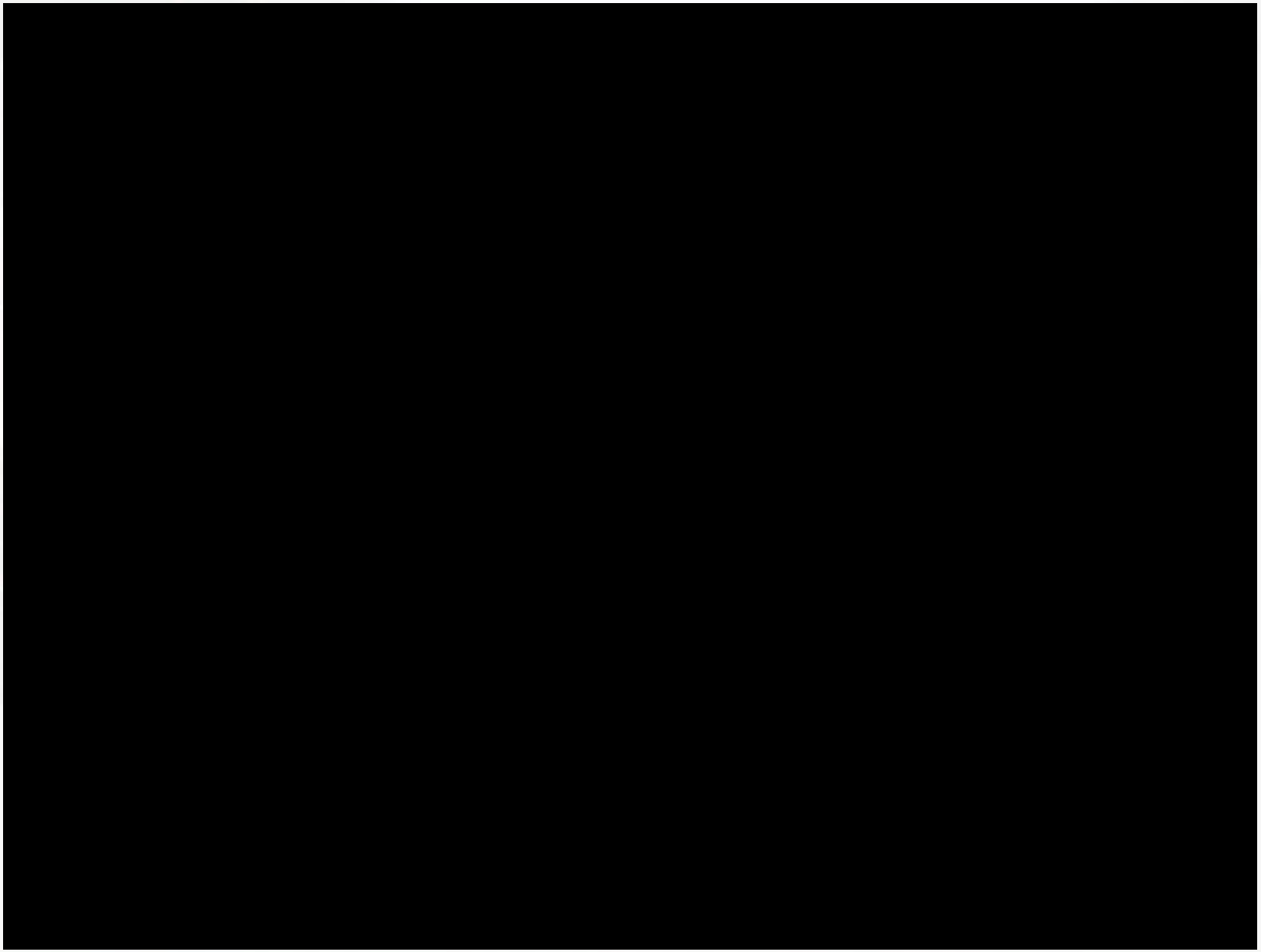
- Für manche Plugins bis heute nicht klar, ob oder was sie tun
- Flüchtigkeitsfehler (z.B. in 1.219 bei gleichen Build-Verfahren und Build-Skript aber unterschiedlichen Targets)
- Wenn Hudson und Plugins verschiedene Versionen der gleichen Library verwenden wollen (z.B. commons-lang)
- Umgang mit Passwörtern in Plugins (z.B. Google Cal)
- Schreiben von Plugins ist nicht einfach (Einarbeitung in Jelly, nur mit Maven)

Hudson
Gemachte Erfahrungen

Positiv aufgefallen ☺

- Sehr schnell „Up and Running“ (~ 1 Stunde)
- Eine Oberfläche für Konfiguration und Betrieb
- Permanente Links (z.B. für LATEST builds)
- Nachträgliches Tagging
- Berechtigungskonzept
- Starke Skalierungsunterstützung
- Schnelle Bearbeitung von Fehlern und Feature Wünschen
- Unzählige Plugins (~ 135)

Ein kleiner Film zum Schluss ...



Literatur und Links

Bücher

- Continuous Integration, Paul M.Duval

Links

- https://hudson.dev.java.net/
- http://www.martinfowler.com/articles/continuousIntegration.html
- http://www.slideshare.net/drluckyspin/continuous-integration
- http://www.slideshare.net/carlo.bonamico/continuous-integration-with-hudson/
- http://jboss-qa.blogspot.com/2007/10/taking-continuous-integration-to.html
- http://www.it-agile.de/build-flashfilm.html
- http://www.sonatype.com/people/2009/01/the-hudson-build-farm-experience-volume-i/ (sowie deren Fortsetzungen)

Literatur und Links

Bilder Referenzen (cc creative commons)

- http://www.flickr.com/photos/carstingaxion/1103931822/
- http://www.flickr.com/photos/lukepdq/99418297/
- http://creativecommons.org/licenses/by-nc-nd/2.0/deed.en
- http://www.flickr.com/photos/paopix/2413495787/
- http://www.flickr.com/photos/sonosalvo/171714927/
- http://www.flickr.com/photos/heyjules/2144592427/
- http://www.flickr.com/photos/destinme/1267500829
- http://www.flickr.com/photos/dullhunk/359634390/
- http://www.flickr.com/photos/k9/556002530/
- http://www.flickr.com/photos/ppdigital/2329376071/
- http://www.flickr.com/photos/iko/739595/

Noch Fragen? Aber gern

Martin Heider arbeitet seit mehr als 15 Jahren im Bereich Software-Entwicklung. Als Freiberufler unterstützt er Kunden in verschiedenen Rollen als SW-Entwickler, SW-Architekt, Testmanager, Team- und Entwicklungsleiter oder Coach.

Seine Erfahrung umfasst international verteilte Projekte sowie Teams verschiedener Größen. Sein besonderes Interesse gilt agilen Methoden und der Herausforderung Software-Entwicklung einfach zu machen, damit alle Beteiligten mit mehr Spaß bessere Ergebnisse erzielen.

Sie erreichen mich unter: mh@infomar.de