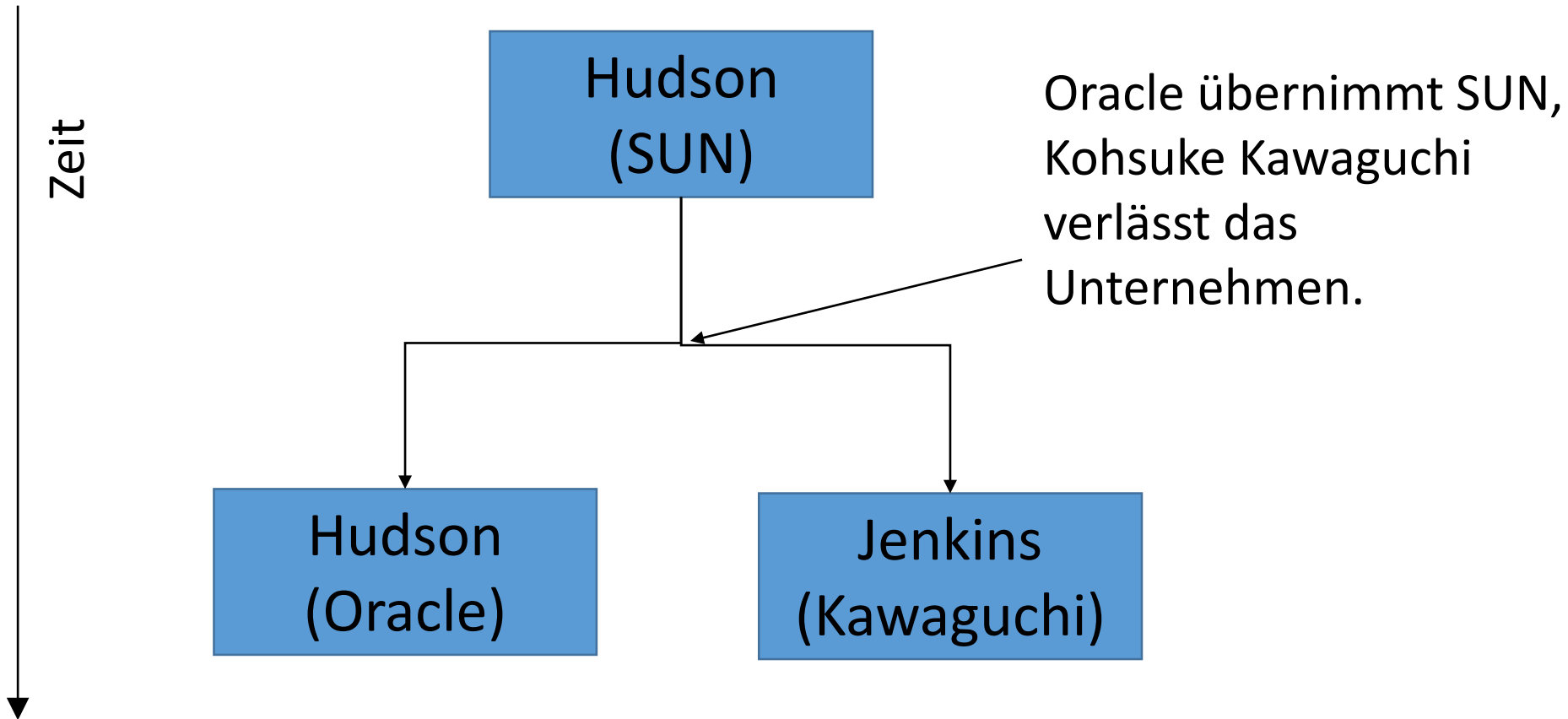


Quellen

- <http://www.eclipse.org/hudson/>
- http://wiki.eclipse.org/The_Hudson_Book
- G. Popp: Konfigurationsmanagement
- <http://hudson-ci.org/>

Hudson vs Jenkins



Beide Server werden aktiv weiterentwickelt.

- Default support für CVS und SVN
- Build Automatisierung
 - Definition des Builds
 - Delegieren an eine build engine (make, ant, maven, custom script, ...)
 - Durchführen: Zeitgesteuert oder bei jeden commit oder manuell per http request
- Automatisiertes Deployment in die Produktionsumgebung
- Testautomatisierung
 - Xunit
 - Funktionale Tests (mit FIT, Selenium, Watir, ...)
- General Purpose Scheduler

Warum ein CI Server?

Entwickler Build

vs

Integrationsbuild

Intention:

- Durchführung auf lokalem Entwicklerrechner
- Schnelles unkompliziertes Kompilieren
- Durchführen der Modultests
- Evtl: Erstellen des Artefakts

➔ Durchführung mit der IDE auf Basis des POMs

Intention:

- Feststellen, in welcher Verfassung sich das Projekt befindet.
- Zentrale Rolle im Projekt
- Kennzeichnung und Auslieferung des Produkts
- Abgleich mit dem Repository

➔ nicht alleine mit Maven möglich ➔ Verwendung eines CI Servers

- Anlegen eines build Profils im Maven POM für den Integrationsbuild.
- Anpassen dieses Builds:
 - Z.B. Kompilieren ohne Debug Informationen zu erzeugen
 - Erzeugen von build Nummern → z.B. mithilfe des buildnumber Plugins (aus dem Mojo Projekt)
 - Ausliefern des Produkts in ein maven repository mithilfe des maven deploy plugins
 - Ggfs weitere Anpassungen
- Abgleich mit dem SVN repository

Warum ein CI Server?

Integrationsbuild als Dreh und Angelpunkt der Qualitätssicherung

- Bei jedem Commit durchführen
- Incl. Code Metriken
- Incl Tests
- Incl. Testmetriken
- Incl Auslieferung.
- Build von Grund auf

Hudson – der Start

- Installation
 - Download
 - Installation in eine Servlet Engine
- Start
 - Definition der bereits zu Beginn benötigten Plugins
 - Maven 3 Build Plugin
 - Hudson Subversion Plugin
 - Systemparameter setzen
 - Maven Installationen
 - E-Mail Benachrichtigungen

Hudson Initial Setup

Hudson CI Server Initial Setup

Core Compatibility Plugins

These core plugins provide key extensions to Hudson to ensure maximum compatibility with a wide range of 3rd party plugins and Operating Systems function. Although these plugins are not required we *strongly* recommend that you install them to ensure maximum compatibility with the existing range of 3rd party Hudson plugins.

<input checked="" type="checkbox"/>	Hudson BIRT Charts Plugin This Plugin provides graphing support to Hudson using BIRT Chart Engine.	3.0.3
<input checked="" type="checkbox"/>	JNA Native Support Plugin Plugin provides support for Native Access using JNA library.	3.0.4
<input checked="" type="checkbox"/>	XPath Provider Plugin XPath Service Provider for Jelly	1.0.2

Featured Plugins

Following are featured plugins. They are tested and certified by Hudson QA team.

<input type="checkbox"/>	Hudson CVS Plug-in Integrates Hudson with CVS SCM	2.2.0
<input type="checkbox"/>	Hudson GIT plugin Integrates Hudson with GIT SCM	2.2.14
<input type="checkbox"/>	Groovy Support Plugin	3.0.3
<input checked="" type="checkbox"/>	Hudson Maven3 Plugin This plug-in adds Maven3 support to Hudson. It adds a builder to Freestyle Project to build maven projects.	3.0.4

Hudson Startseite



Hudson

[Hudson](#)

-  [Neuen Job anlegen](#)
-  [Hudson verwalten](#)
-  [Benutzer](#)
-  [Build-Verlauf](#)
-  [New View](#)

Geplante Builds
Keine Builds geplant

Build-Prozessor Status
Master 0/2
Idle



Jobs Status

Willkommen bei Hudson! Legen Sie einen [neuen Job](#) an, um loszulegen.

Explizite Maven Installation angeben

Maven

Maven Installationen

Maven

Name

apache-maven-3.3.3

MAVEN_HOME

D:\OTH\Vorlesungen\MST-SS2015\mavenBspe\apache-maven-3.3.3



Automatisch installieren



Maven entfernen

Maven hinzufügen

Liste der Maven Installationen auf diesem System

E-Mail Server Konfigurieren

E-Mail Benachrichtigung

SMTP-Server



Standardendung für E-Mail-Adressen



E-Mail-Adresse des Systemadministrators

Adresse nicht konfiguriert <nobody@nowhere>



Hudson URL



Advanced...

- **Projekt Anlegen → in Hudson „Neuen Job anlegen“**
- **Projektkonfigurationen anlegen**
 - Repository Typ → Subversion
 - URL des svn servers/trunk
 - Checkout strategy: svn update
- **Zeitgesteuerte Ausführung des Integrationsbuilds festlegen**
 - Z.B.: @hourly
 - Alternative: *Source Code Management System abfragen*

Hudson – neuen Job anlegen



The screenshot shows the Hudson web interface. The top header is purple with the Hudson logo (a cartoon character) and the word "Hudson". Below the header, on the left, is a sidebar with navigation links: "Neuen Job anlegen" (with a folder icon), "Hudson verwalten" (with a wrench icon), "Benutzer" (with a group of people icon), "Build-Verlauf" (with a document icon), and "New View" (with a folder icon). Below these links are two status boxes. The first, "Geplante Builds", shows "Keine Builds geplant". The second, "Build-Prozessor Status", shows "Master 0/2" and "Idle". On the right side of the interface, the title "Jobs Status" is displayed with a folder icon. Below the title, a message reads: "Willkommen bei Hudson! Legen Sie einen [neuen Job](#) an, um loszulegen."

Hudson

[Neuen Job anlegen](#)

[Hudson verwalten](#)

[Benutzer](#)

[Build-Verlauf](#)

[New View](#)

Geplante Builds

Keine Builds geplant

Build-Prozessor Status

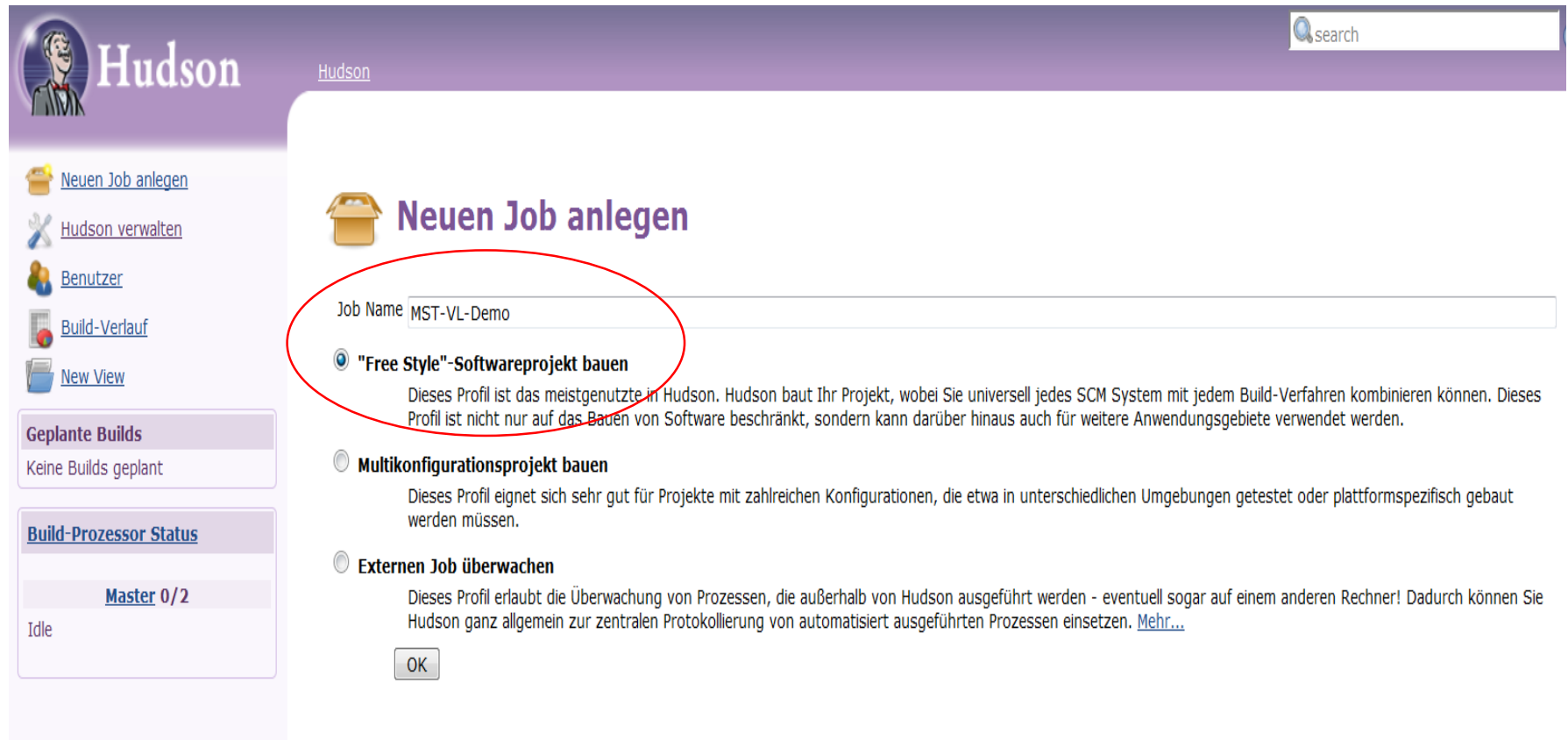
Master 0/2

Idle

Jobs Status

Willkommen bei Hudson! Legen Sie einen [neuen Job](#) an, um loszulegen.

Hudson – neuen Job anlegen



The screenshot shows the Hudson web interface. On the left is a sidebar with navigation links: [Neuen Job anlegen](#), [Hudson verwalten](#), [Benutzer](#), [Build-Verlauf](#), and [New View](#). Below these are sections for **Geplante Builds** (Keine Builds geplant) and **Build-Prozessor Status** (Master 0/2, Idle). The main content area is titled **Neuen Job anlegen** and features a search bar at the top right. A red circle highlights the **Job Name** input field containing "MST-VL-Demo" and the **"Free Style"-Softwareprojekt bauen** radio button option. Below this, three other options are listed: **Multikonfigurationsprojekt bauen** and **Externen Job überwachen**, each with a descriptive paragraph. An **OK** button is at the bottom.

Hudson

Neuen Job anlegen

Job Name

☒ **"Free Style"-Softwareprojekt bauen**
Dieses Profil ist das meistgenutzte in Hudson. Hudson baut Ihr Projekt, wobei Sie universell jedes SCM System mit jedem Build-Verfahren kombinieren können. Dieses Profil ist nicht nur auf das Bauen von Software beschränkt, sondern kann darüber hinaus auch für weitere Anwendungsgebiete verwendet werden.

☐ **Multikonfigurationsprojekt bauen**
Dieses Profil eignet sich sehr gut für Projekte mit zahlreichen Konfigurationen, die etwa in unterschiedlichen Umgebungen getestet oder plattformspezifisch gebaut werden müssen.

☐ **Externen Job überwachen**
Dieses Profil erlaubt die Überwachung von Prozessen, die außerhalb von Hudson ausgeführt werden - eventuell sogar auf einem anderen Rechner! Dadurch können Sie Hudson ganz allgemein zur zentralen Protokollierung von automatisiert ausgeführten Prozessen einsetzen. [Mehr...](#)

■ Buildverfahren

- Build Schritt anlegen
- Variante *Invoke Maven 3*
- Parameter für den Aufruf von Maven angeben
z.B.: *mvn clean deploy -Pbuild int*

■ Benachrichtigungen per email:

- Post-Build Aktionen
 - Emails bei jedem instabilen Build senden
 - Ggfs an den Entwickler des letzten Changesets

Hudson – Build Schritt

[Zurück zur Übersicht](#)

[Status](#)

[Änderungen](#)


[Jetzt bauen](#)

[Löschen](#)

[Konfigurieren](#)

Build-Verlauf ([Trend](#))

[Alle Builds](#) [Nur Fehlschläge](#)



Job Configurations

Projektname

Beschreibung

☐ Alte Builds verwerfen

☐ Dieser Build ist parametrisiert.

☐ Projekt deaktivieren (Es werden keine weiteren Builds ausgeführt, bis das Projekt wieder reaktiviert wird.)

☐ Parallele Builds ausführen, wenn notwendig

Erweiterte Projekteinstellungen

Source-Code-Management

☒ Keines

☐ Subversion

Build-Auslöser

☐ Starte Build, nachdem andere Projekte gebaut wurden.

☐ Builds zeitgesteuert starten

☐ Source Code Management Sy:

☐ Build when Maven dependenci

Buildverfahren

Build-Schritt hinzufügen ▼

Buildverfahren

Build-Schritt hinzufügen ▼

- Shell ausführen
- Maven 2 (Legacy) aufrufen
- Ant aufrufen
- Windows Batch Datei ausführen
- Invoke Maven 3

Post-Build-Aktionen

Hudson – Build Schritt

Build-Auslöser

☐ Starte Build, nachdem andere Projekte gebaut wurde

☒ Builds zeitgesteuert starten


Zeitplan

@hourly

☐ Source Code Management System abfragen

☐ Build when Maven dependencies have been updated

Buildverfahren

 **Invoke Maven 3**

Maven 3

Goals

Properties

Hudson – Build Schritt

Buildverfahren

Invoke Maven 3

Maven 3

Goals

Properties

Build-Schritt hinzufügen ▼

Post-Build-Aktionen

☐ Nachgelagerte Testergebnisse zusammenfassen

☐ Fingerabdrücke von Dateien aufzeichnen, um deren Verwendung zu verfol

☒ Veröffentliche JUnit-Testergebnisse.

Testberichte in XML-Format

Es sind reguläre Ausdrücke wie z.B. 'myproject/target/test-reports'.
Ausgangsverzeichnis ist der [Arbeitsbereich](#).

☐ Retain long standard output/error

☐ Aufschaltbare Ausgabedateien

- **Veröffentlichen der JUnit Testergebnisse**
 - Option im Abschnitt Post-Build-Aktionen
 - → Dateinamenpattern für die Testergebnisse:
`**/surefire-reports/*.xml`

- **→ Fertig:** Jetzt wird stündlich ein Integrationsbuild gefahren. Der Status ist in der Hudson GUI sichtbar. Bei fehlgeschlagenen Builds werden die Entwickler per email informiert und die Testergebnisse sind einsehbar.

Hudson Build Verlauf

 [Zurück zur Übersicht](#)

 [**Status**](#)

 [Änderungen](#)

 [Jetzt bauen](#)

 [Löschen](#)

 [Konfigurieren](#)

Build-Verlauf

([Trend](#))

	#2	29.06.2015 16:37:32	
	#1	29.06.2015 16:37:24	

 [Alle Builds](#)
 [Nur Fehlschläge](#)

- Checkstyle
(<http://checkstyle.sourceforge.net/>)
Tool, um Einhaltung von Coding Standards zu überprüfen
- Integration in den build
 - Checkstyle konfigurieren: Datei Checkstyle-config.xml im Wurzel Verzeichnis des trunk
 - Verwendung des maven-checkstyle-plugin und Einbinden in die Erstellung der Projekt Homepage
- Analog mit Code Coverage Tools

Weitere CI Server

Siehe z.B. die Liste unter

http://de.wikipedia.org/wiki/Kontinuierliche_Integration#Software