Individuelle IT-Konzepte
und Softwarelösungen

Gesellschaft für
Informations- und
Kommunikationssysteme mbH

Softwaretests

Werkzeuge zur Automatisierung

für

Thementag „Wer testet, ist feige“

24.06.2009

Autor:
Markus Alvermann

Agenda

- Motivation
- Versionsverwaltung
- Build-Tools
- Unit-Tests
- GUI-Tests
- Continuous Integration Server
- Defect-Tracking-Tools
- Fazit

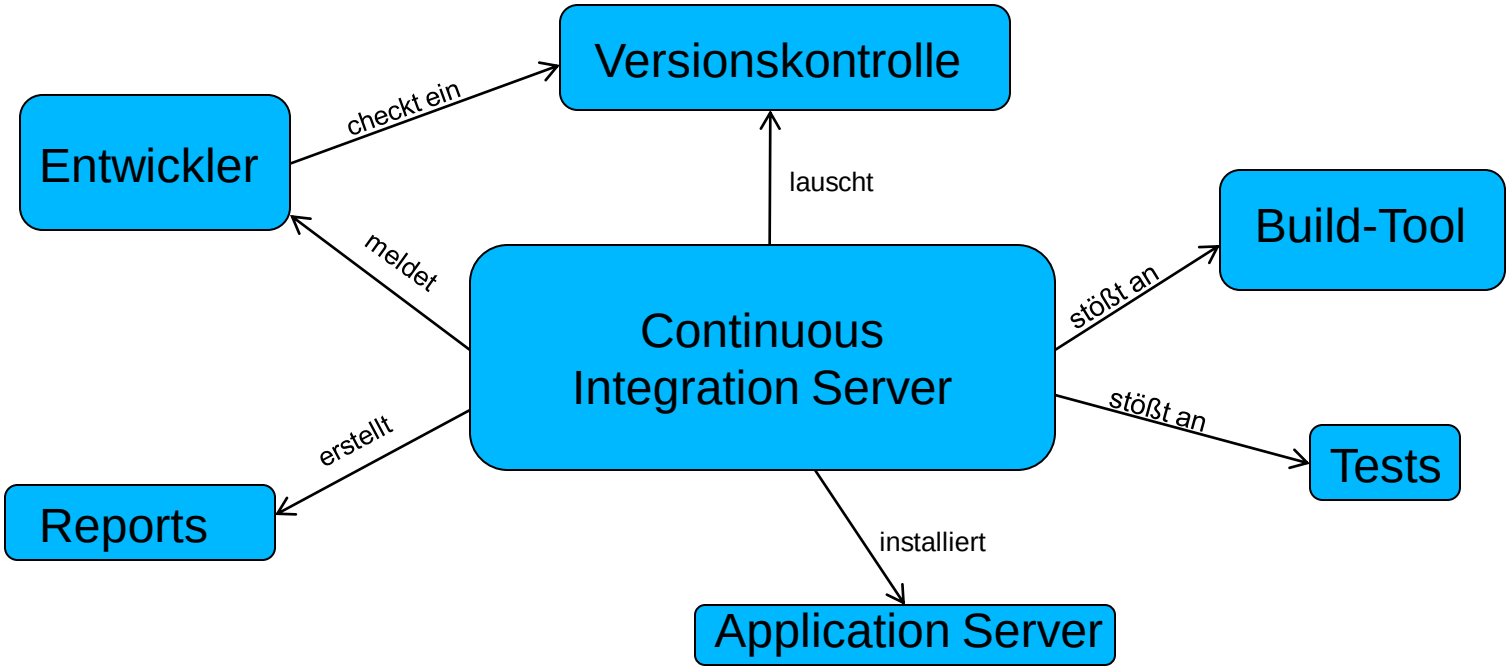
Wo sind wir

- **Motivation**
- Versionsverwaltung
- Build-Tools
- Unit-Tests
- GUI-Tests
- Continuous Integration Server
- Defect-Tracking-Tools
- Fazit

Motivation I

- Ziel: Automatisiertes Testen
- Aufwandsreduzierung
- Homogene Testumgebung
- Frühes Erkennen von Fehlern
- Mögliche Lösung: Continuous Integration

Motivation II



Wo sind wir

- Motivation
- **Versionsverwaltung**
- Build-Tools
- Unit-Tests
- GUI-Tests
- Continuous Integration Server
- Defect-Tracking-Tools
- Fazit

Versionsverwaltung: CVS vs. SVN I

- Concurrent Versions System (CVS)
- Subversion (SVN)
- Server-Client-Architekturen
- Verwalten der Änderungen
- Rechtesystem
- Löschen von Verzeichnissen
- Verschieben von Dateien

Versionsverwaltung: CVS vs. SVN II

- **Branch-Konzept**
- **Check-In, Check-Out**
- **Fazit: SVN beseitigt viele Schwächen von CVS**

Wo sind wir

- Motivation
- Versionsverwaltung
- **Build-Tools**
- Unit-Tests
- GUI-Tests
- Continuous Integration Server
- Defect-Tracking-Tools
- Fazit

Build-Tools – Ant vs. Maven2 I

- Tools für die automatische Erstellung von Software
- Steuerung durch xml-Dateien
- Maven2: Convention over Configuration
- Tasks vs. Plugins
- Auflösen von Abhängigkeiten
- Fazit

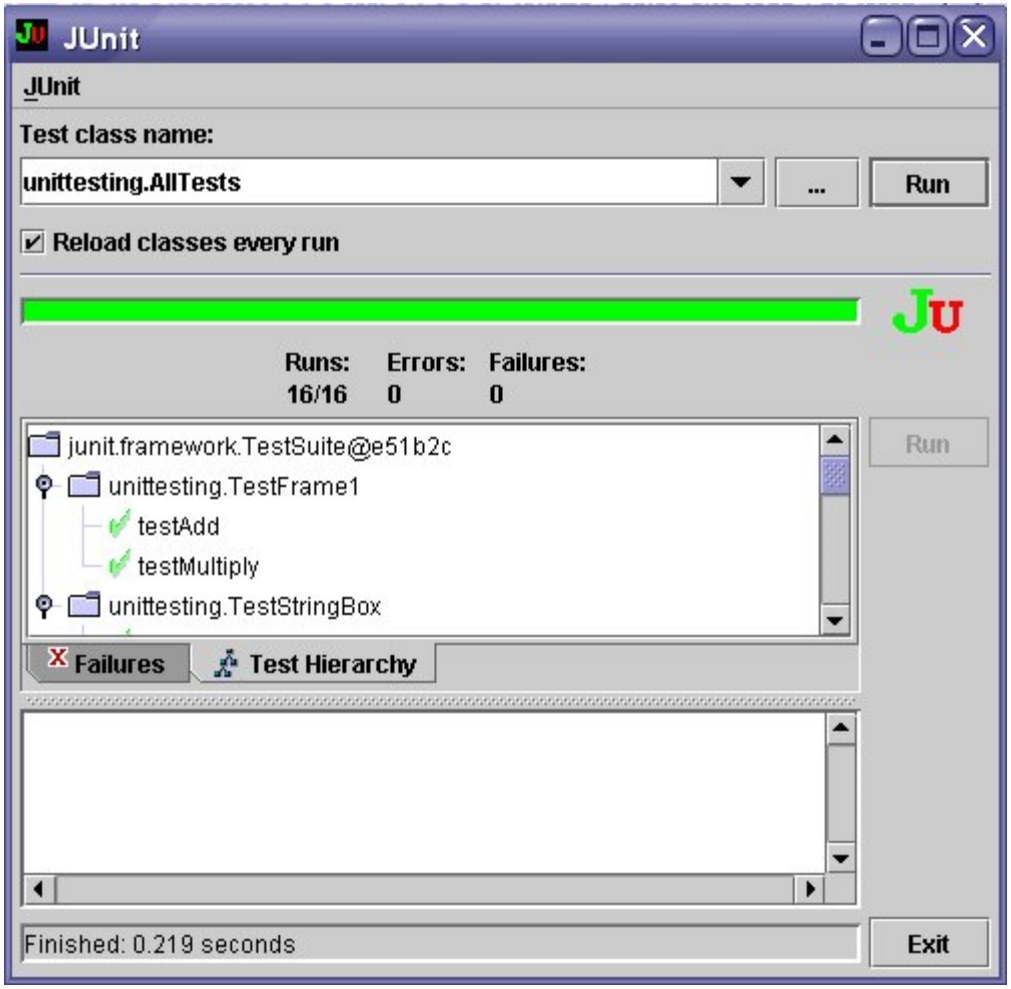
Wo sind wir

- Motivation
- Versionsverwaltung
- Build-Tools
- **Unit-Tests**
- GUI-Tests
- Continuous Integration Server
- Defect-Tracking-Tools
- Fazit

Unittests – JUnit I

- **Framework für automatisierte Unittests von Java-Programmen**
- **Testgetriebene Entwicklung („Extreme Testing“)**
  - Vorgehen: Zuerst die Testklasse schreiben, dann die Klasse gegen die getestet werden soll
- **Tests sind in Java programmierte Klassen**
- **Definition per Annotations**
- **Tests können zu Testsuiten zusammengefasst werden**
- **Zwei Testergebnisse: grün und rot**
- **Reporterstellung**

Unittests – JUnit II



Unittests - EasyMock

- Motivation
- Programmbibliothek zum Erstellen von Mock-Objekten
- Isolation von Units durch Mock-Objekte
- Mock-Objekt-Erzeugung automatisch und dynamisch zur Laufzeit
- Reihenfolge
- Methoden: expect, replay und verify
- Typische Mock-Objekte

Wo sind wir

- Motivation
- Versionsverwaltung
- Build-Tools
- Unit-Tests
- **GUI-Tests**
- Continuous Integration Server
- Defect-Tracking-Tools
- Fazit

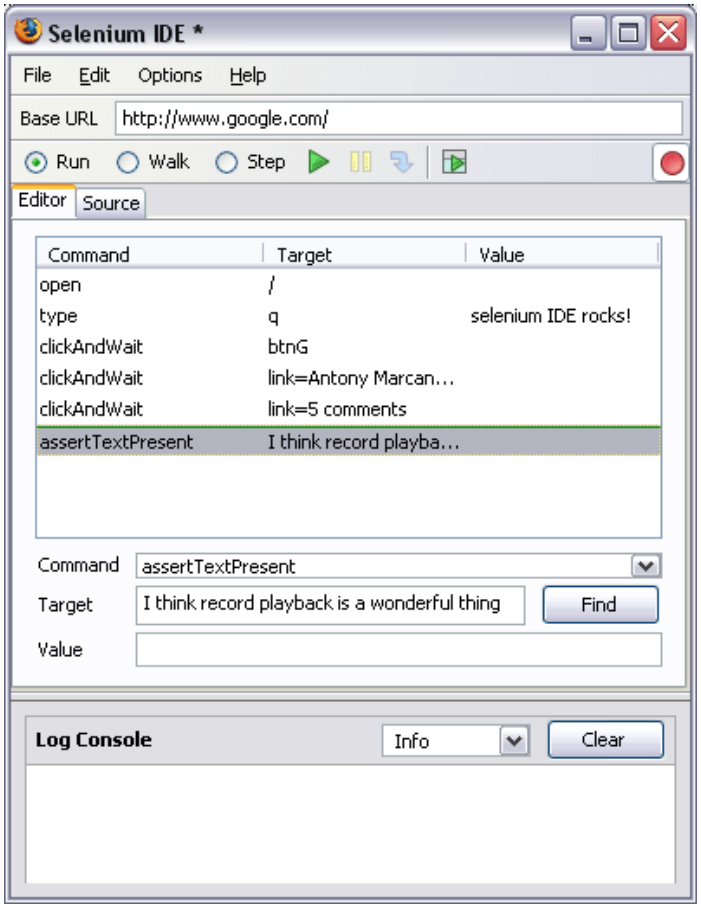
GUI-Tests- Einführung

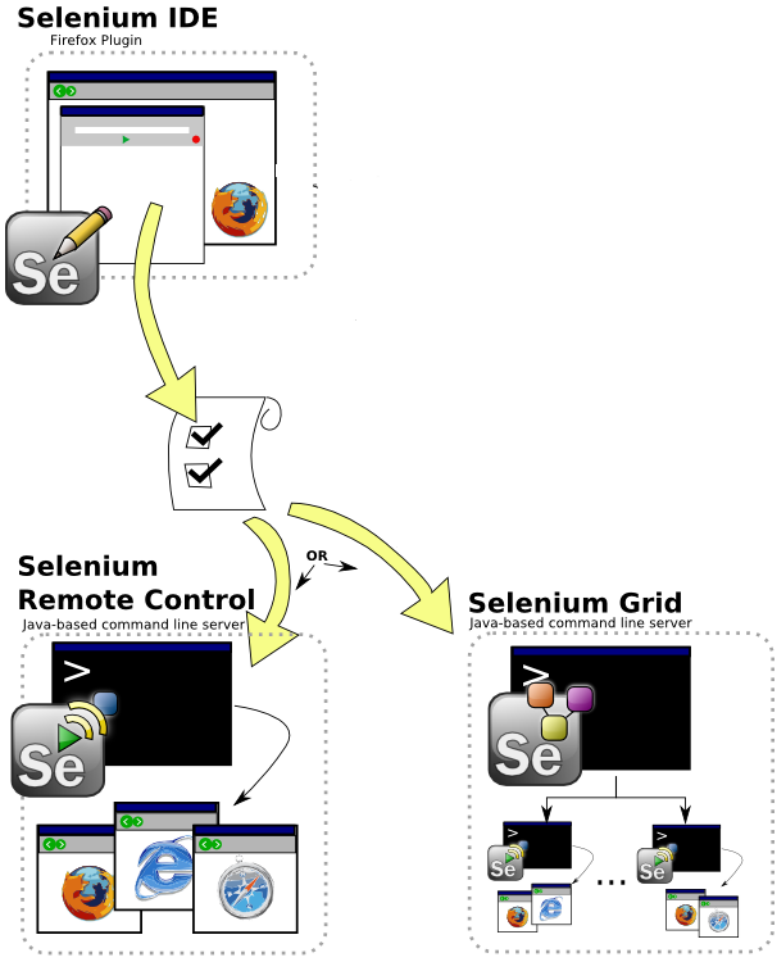
- **Ziel: Automatisiertes Testen von Graphical User Interfaces (kurz: GUI)**
- **Standalone-Applikationen (bspw. Swing, Applets) und Webapplikationen**
- **Aufnehmen von Tests vs. Programmieren von Tests**
- **Was wird getestet:**
  - Formulareingaben
  - Funktioniert der Workflow?
  - Greifen die Validierungen?
  - Kompatibilität (Webapplikationen)

GUI-Tests- Selenium I

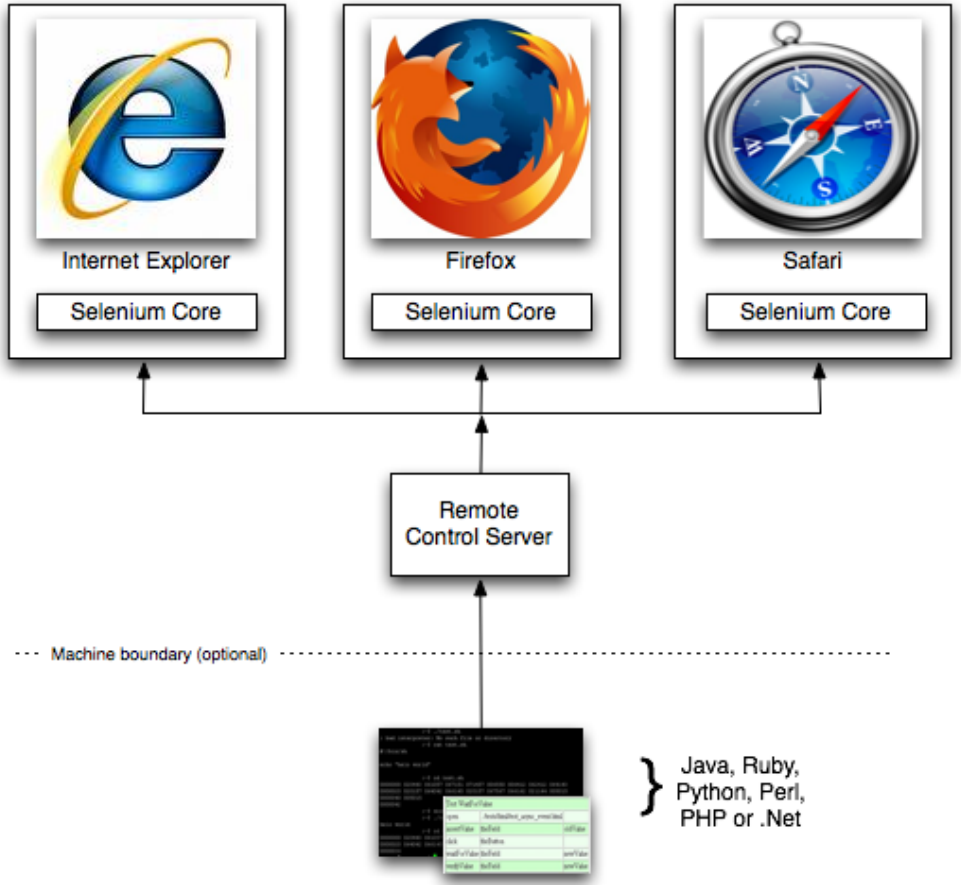
- **Testframework für Webapplikationen**
- **Drei Komponenten:**
  - Selenium IDE
  - Selenium Remote Control (RC)
  - Selenium Grid
- **Aufzeichnen von Scripts mit Selenium IDE**
- **Exportieren von aufgezeichneten Scripts und Ausführen mit Selenium RC oder Selenium Grid**

GUI-Tests- Selenium II





GUI-Tests- Selenium III



GUI-Tests- Jemmy I

- **Java-Bibliothek zum Testen von Swing- und AWT-Applikationen und Applets**
- **Tests werden in Java geschrieben und können dann bspw. mit JUnit ausgeführt werden**
- **Test läuft in der gleichen JVM wie die Applikation**
- **Testablauf**
  - Applikation starten
  - Fenster finden
  - Komponente finden und Aktionen auf dieser Komponente ausführen

GUI-Tests- Jemmy II

- **Kurzer Beispielcode:**

```
import org.netbeans.jemmy.*;
import org.netbeans.jemmy.explorer.*;
import org.netbeans.jemmy.operators.*;

public class WaitWindowSample implements Scenario {
    public int runIt(Object param) {
        try {
            //start application
            new ClassReference("org.netbeans.jemmy.explorer.GUIBrowser").startApplication();
            //wait frame
            new JFrameOperator("GUI Browser");
        } catch(Exception e) {
            e.printStackTrace();
            return(1);
        }
        return(0);
    }
    public static void main(String[] argv) {
        String[] params = {"WaitWindowSample"};
        org.netbeans.jemmy.Test.main(params);
    }
}
```

Wo sind wir

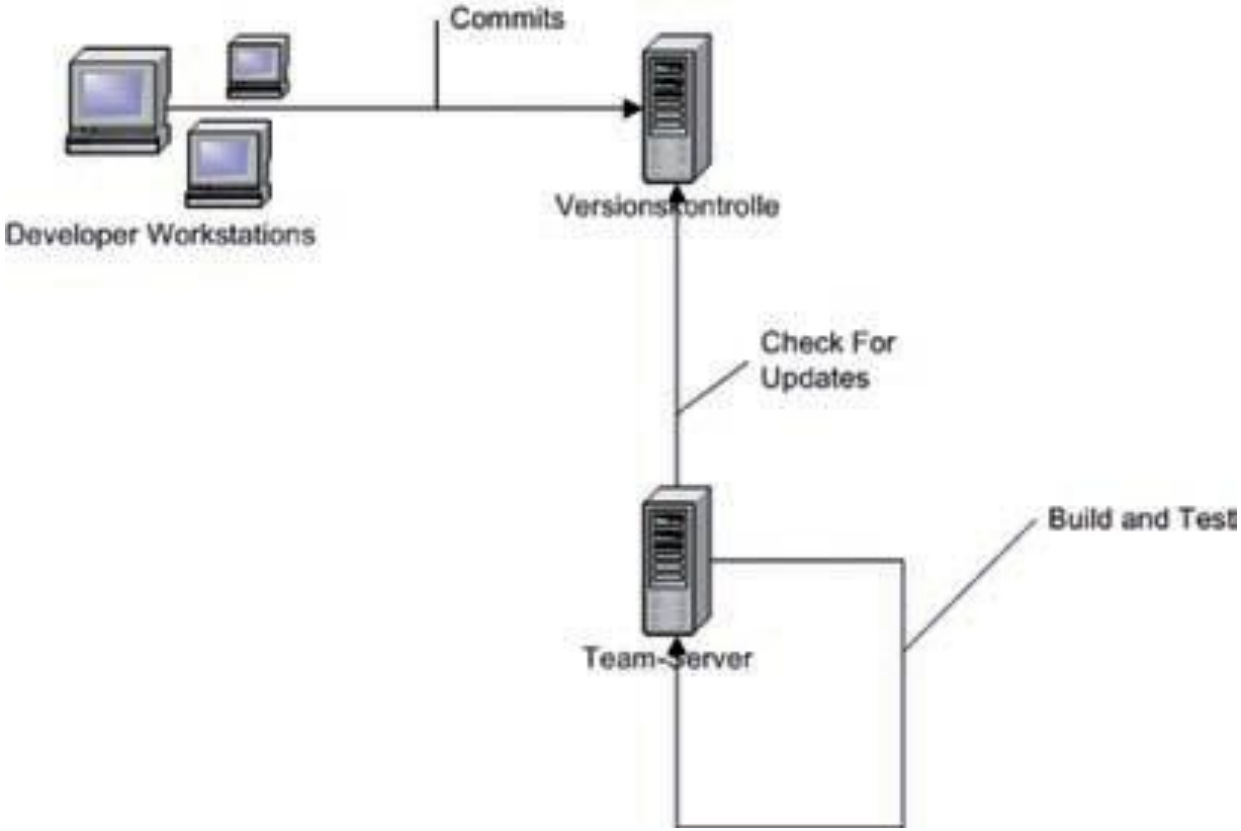
- Motivation
- Versionsverwaltung
- Build-Tools
- Unit-Tests
- GUI-Tests
- **Continuous Integration Server**
- Defect-Tracking-Tools
- Fazit

Continuous Integration Server - Einführung

- **Applikation, welche die kontinuierliche Integration von Software vereinfacht und automatisiert**
- **Regelmäßige Neu-builden und Testen einer Software**
- **Zentrale Stelle für die Integration von Quellcode**
- **Der CIS entscheidet, ob eine Integration erfolgreich war**
- **Nur sinnvoll, wenn der Build zu 100% automatisiert abläuft (Kompilieren, Verpacken, Testen)**
- **Weitere Aufgaben: Erzeugen von Reports, Benachrichtigungen versenden, Snapshots bilden, Integrations- & Akzeptanztests auf einer produktionsähnlichen Umgebung**

Continuous Integration Server - Einführung

- **Voraussetzung: Softwareprojekt ist in einem zentralen Repository abgelegt**
- **Weiteres Vorgehen (bspw. Neu-Builden, Testen, Installieren) einstellbar**



Continuous Integration Server – Hudson I

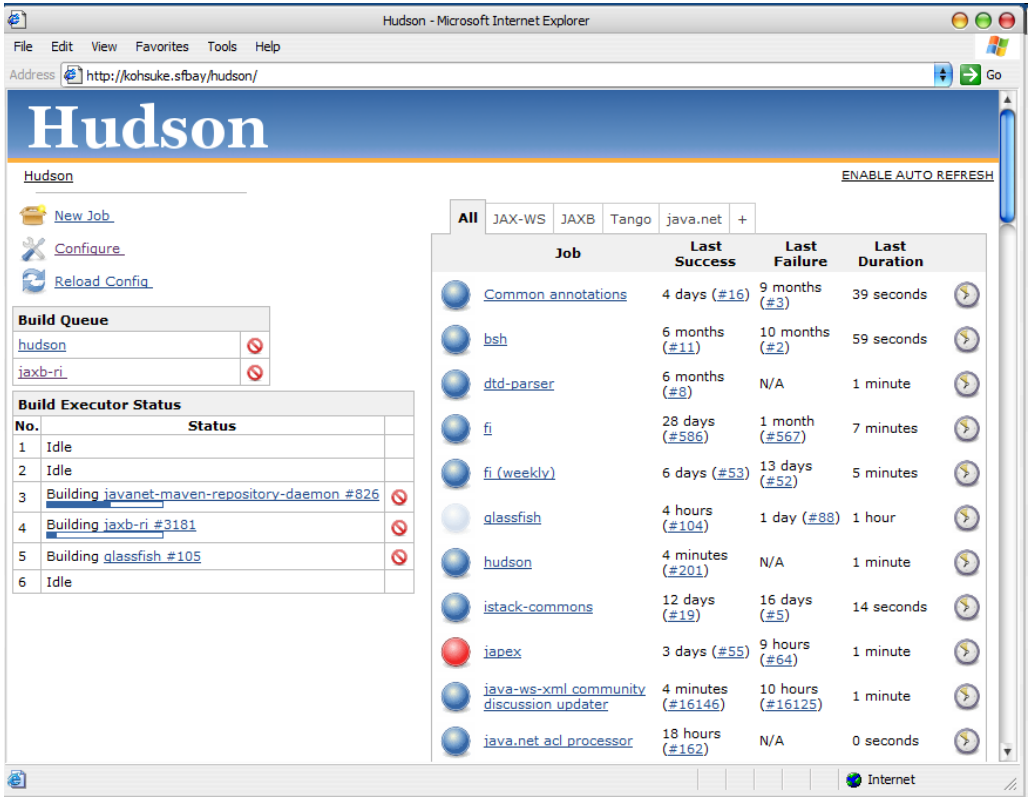
- **In Java geschrieben, läuft in einem Servlet-Container**
- **Unterstützung im Lieferumfang**
  - Quellcodeverwaltungen wie Subversion und CVS
  - Build-Tools wie Ant und Maven
  - Unittest-Tools wie JUnit und TestNG
  - Reporterstellung
  - Benachrichtigung per RSS, email, IM

Hudson

- **Einfache Installation**
- **Einfache Konfiguration über Web-GUI (per Assistent)**
- **Zeitpunkt für neue Builds anhand verschiedener Parameter möglich**

Continuous Integration Server – Hudson II

- **Durch Plugins erweiterbar (selbstgeschrieben vs. zur Verfügung stehende)**
- **Benutzerverwaltung über Plugin (DB-basiert)**



Continuous Integration Server – TeamCity I

- Java-Applikation für Servlet-Container
- Entwickelt von JetBrains, daher sehr gute Integration in IntelliJIdea
- Kommerzielle Lizenz, aber freie Lizenz mit einigen Einschränkungen erhältlich
- Konfiguration von Builds im Server, Ausführung in einem Build-Agenten
- Toolunterstützung: diverse Versionsverwaltungen, diverse Build-Tools, .NET-Unterstützung, umfangreiche Statistiken, Unterstützung mehrerer JDKs in einem Projekt, Benachrichtigungen, Reports, Quellcodeanalyse

Continuous Integration Server – TeamCity II

- **Konfiguration von Projekten in 2 Schritten:**
  - Projekt anlegen (Name, Beschreibung, Build-Nummerierung)
  - Projekt konfigurieren
    - Versionsverwaltungsinformationen
    - Build-Runner
    - Build-Trigger
    - Projekt-Abhängigkeiten
    - Umgebungsvariablen und System-Properties
    - Anforderungen an die Agents

Wo sind wir

- Motivation
- Versionsverwaltung
- Build-Tools
- Unit-Tests
- GUI-Tests
- Continuous Integration Server
- **Defect-Tracking-Tools**
- Fazit

Defect-Tracking-Tools - Einführung

- Erfassen von Programmfehlern
- Qualitätssicherung
- Verbesserung der Kommunikation
- Zentrales Archiv
- Zuständigkeiten
- Übersicht über den Gesamtzustand eines Projektes

Defect-Tracking-Tools – Mantis I

- Bugtracker und Feature-Request-Werkzeug
- Basiert auf PHP
- Server-Client-Architektur
- Einfache Konfiguration
- Einteilung in Projekte und Unterprojekte
- Ebenenbasierte Zugriffsrechte
- Fehler in Problemberichten
- Datenverwaltung in Datenbank
- Filter

Defect-Tracking-Tools – Mantis II

Recently Modified [^] (1 - 10 / 2828)	
0006141	Fret Buzz on the 12th Fret D String [Demo] Website - **2009-06-14 12:32**
0006148	this is just another test [Demo] GUI - 2009-06-14 07:59
0006150	Board problem [Demo] Other - 2009-06-14 07:55
0006149	Initial RFP and Selection for Longitudinal Study [Demo] Other - 2009-06-13 10:03
0006147 ^	Demo [Demo] Website - 2009-06-13 08:10
0006145 ^^	afsaf [Demo] Website - 2009-06-13 00:06
0006146	this is something [Demo] GUI - 2009-06-12 19:03
0006144	asdfasdf [Demo] GUI - 2009-06-12 15:49
0006136	Schnittstelle Kaffeemaschine <-> SoftM entwickeln [Demo] Other - 2009-06-12 10:06
0006129	test [Demo] GUI - 2009-06-12 04:46

Defect-Tracking-Tools – Mantis III

Viewing Issue Advanced Details [Jump to Notes] [Wiki]

ID	Category	Severity	Reproducibility
0006141	[Demo] Website	major	always

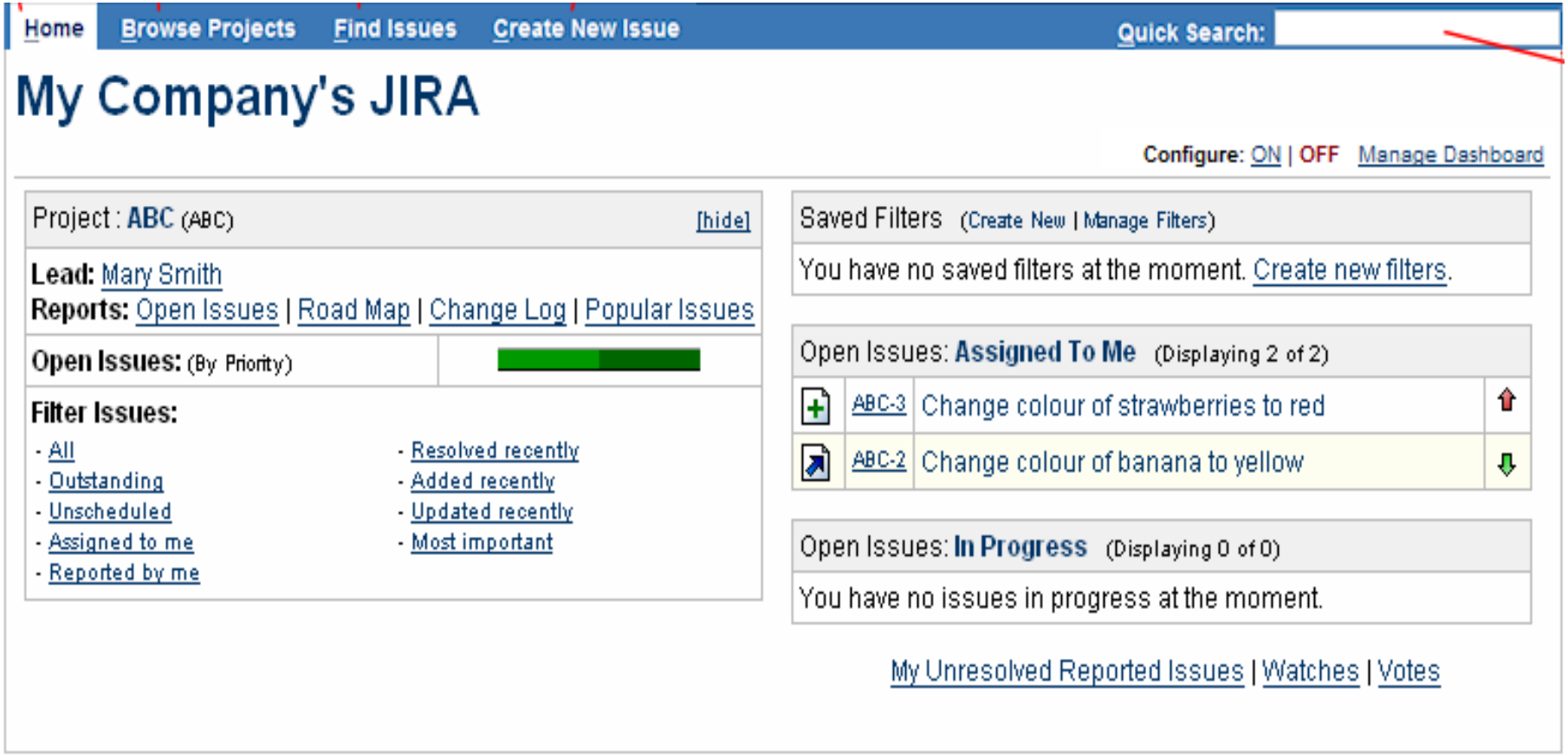
Reporter	cockroach	**View Status**	public
Assigned To	laklak		
Priority	none	**Resolution**	open
Status	assigned		
Projection	none		
ETA	none	**Fixed in Version**	
		Target Version	

Summary	0006141: Fret Buzz on the 12th Fret D String
Description	My mandolin sounds great! I got it real cheap, but it buzzes on the 12th fret of the 2 doubled recourse D strings.
Steps To Reproduce	
Additional Information	I've already raised the bridge with the two thumbscrews. It doesn't bother me too much because I don't go that high up usually.
Tags	No tags attached.
Attached Files	

Defect-Tracking-Tools – Jira I

- Bugtracker und Projekt-Management-Werkzeug
- Atlassian
- J2EE
- Server-Client-Architektur
- Hohe Flexibilität
- Erweiterbar
- Filter
- Personalisierte Startseite (Dashboard)
- Einstellbarer Workflow

Defect-Tracking-Tools – Jira II



Wo sind wir

- Motivation
- Versionsverwaltung
- Build-Tools
- Unit-Tests
- GUI-Tests
- Continuous Integration Server
- Defect-Tracking-Tools
- **Fazit**

Fazit

- Vorstellung gängiger Tools
- Auswahl der geeigneten Werkzeuge
- Automatisiertes Testen führt in der Regel zu einem schnellen Erkennen und Beseitigen von Fehlern
- Um einen CIS effizient einzusetzen, sollten alle in diesem Vortrag vorgestellten Testwerkzeuge verwendet werden, da ansonsten die Testkette unvollständig ist
- Hoher Initialaufwand, langfristig geringerer Wartungs- und Pflegeaufwand

Referenzen

cvs:	**http://www.nongnu.org/cvs/**
svn:	**http://subversion.tigris.org/**
Apache Ant:	**http://ant.apache.org/**
Apache Maven:	**http://maven.apache.org/**
JUnit:	**http://www.junit.org/**
EasyMock:	**http://easymock.org/**
Selenium:	**http://seleniumhq.org/**
Jemmy:	**https://jemmy.dev.java.net/**
Hudson:	**https://hudson.dev.java.net/**
TeamCity:	**http://www.jetbrains.com/teamcity/**
Mantis:	**http://www.mantisbt.org/**
Jira:	**http://www.atlassian.com/software/jira/**

www.iks-gmbh.com