

# Developers' Little Helpers

Ein Überblick über Softwareprojekte die die tägliche Arbeit des Entwicklers vereinfachen.

# Überblick

- IDEs
- Versionierungssysteme
- Build Tools
- Code Analysis / Metrics
- Unit Tests
- Continuous Integration
- Sonstiges



# IDEs

- NetBeans IDE
  - Open Source
  - Support für viele Sprachen
  - für alle Java Bereiche (von GUI bis EE)
- Eclipse
  - Open Source
  - sehr viele Plug-ins
- IntelliJ IDEA
  - Community Edition kostenlos
  - Ultimate Edition für Open Source Projekte kostenlos
- BlueJ
  - besonders für Anfänger und Lehre geeignet

# Code Versionierung

- CVS
  - <http://www.cvshome.org/>
- Subversion
  - <http://subversion.apache.org/>
- Git
  - <http://git-scm.com/>
- Mercurial
  - <http://mercurial.selenic.com/>

# Build Tools

- Ant
  - sehr flexibel
  - Ant-Tasks für viele Tools
  - <http://ant.apache.org/>
- Maven
  - hat einen "Standard-Weg"
  - kann auch Ant-Tasks ausführen
  - <http://maven.apache.org/>



# Code Analysis / Metrics 1

## Prüfung des Code Stils

- Erfüllung von Standards / Konventionen
- Leere Code Blöcke und Statements
- Länge von Methoden und Klassen
- etc.

# Code Analysis / Metrics 2

## Stil Prüfung anhand von Metriken

- Komplexität von Methoden
- Change Risk Anti-Patterns (CRAP)
- Dependency Analyse

# Code Analysis / Metrics 3

## Prüfen auf gängige Fehler

- "Fehler"-Prüfung anhand von Patterns
- Durchsuchen des Quellcodes oder Bytecodes nach "gängigen" Fehlermustern
- Beispiele:
  - Instantiieren von Klassen die nur static Methoden haben
  - Fehlendes Schließen von Datenbank Ressourcen



# Code Analysis - CheckStyle

Analysiert den Stil des Quellcodes

- Hilft Coding Standards einzuhalten
- Commandozeilen-Tool
- Ant Task
- Plug-ins für IDE's
- <http://checkstyle.sourceforge.net/>



# Code Analysis - CRAP4J

Erstellt Berichte über die Code Komplexität

Ant Task

Eclipse Plug-in

<http://www.crap4j.org>

# Code Analysis - FindBugs

(Byte-)Code Analyse anhand von Bug-Patterns

- Kommandozeilen-Tool
- Ant Task
- GUI
- Eclipse Plug-in
- <http://findbugs.sourceforge.net/>



# Code Analysis - PMD

(Source-)Code Analyse anhand von Regelsätzen

- Kommandozeilen-Tool
- Ant Task
- GUI
- Plug-ins / Integration in diverse IDEs und Editoren
- <http://pmd.sourceforge.net/>

# Code Analysis - UCDetector

Sucht ungenutzten (alten) Code

- Eclipse Plug-in
- Ant Task
- <http://www.ucdetector.org/>

# Unit Tests



## Was ist ein Unit Test?

- Code, der die Funktionalität des zu testenden Codes überprüft.
- Kann sicherstellen, dass die Funktionalität durch Codeänderungen erhalten bleibt.

# Unit Tests mit JUnit 4.x

## Erstes Unit Test Framework für Java

- Testmethoden werden einfach mit Annotationen versehen
- Integration in Eclipse
- Ant Task
- Reporting
- <http://www.junit.org>

# Unit Tests mit TestNG

Entstand aufgrund der Einschränkungen in JUnit 3

- setzt an, wo JUnit aufhört
- Plugin für Eclipse, NetBeans, IDEA
- Ant Task
- <http://testng.org>



# Continuous Integration mit Hudson

Continuous Integration automatisiert den Build Prozess

- Browser-basiert
- Ausführen von Tests
- Ausführen von beliebigen Ant Tasks
- Erweiterbar durch 200+ Plug-ins



# Continuous Integration - Alternativen

- Atlassian Bamboo  
<http://www.atlassian.com/software/bamboo/>
- Cruise Control  
<http://cruisecontrol.sourceforge.net/>

# Sonstiges - VisualVM

Visualisiert den Zustand und Prozesse innerhalb der VM

- Thread inspection
- Garbage Collection
- Memory Usage
- Profiler



# Sonstiges - JRebel

Beschleunigt den Entwicklungsprozess

Quellcode Änderung bewirkt den Austausch des Bytecodes zur Laufzeit der VM

dadurch entfallen VM Neustart und Initialisierung der Applikation

# Vergessen aber unbedingt erwähnenswert?

Am Ende des Vortrags aus dem Publikum zusammengetragen

- Bugtracker
- Ivy (als Ergänzung zu Ant) <http://ant.apache.org/ivy/>
- Bookmark- /Ressourcesammlungen (Delicious & Co)