

Testgetriebene Entwicklung, aber womit?

Aufbau einer Testlandschaft mit Open-Source



Themen/Agenda

- **Ein Schauermärchen als Ausgangssituation**
- **Verwendete Testumgebung**
- **Tests und Testautomatisierung**
- **Und so läuft`s ...**
- **Fazit/Zusammenfassung**
- **Links und Quellen**

Ausgangssituation



Ausgangssituation

- **Wenn Sie die Wahl hätten ...**

**„Kraut und Rüben“ Source-Code
mit automatisierten Tests**

oder

1a Source-Code ohne Tests

Ausgangssituation

- **Wünschen wir uns nicht alle folgende Situation?**
 - **Ihre Anwendung ist entwickelt und läuft erfolgreich**
 - **Immer mehr Kunden setzen Ihre Software ein**
 - **Ihre Kunden bestürmen Sie mit neuen Anforderungen**

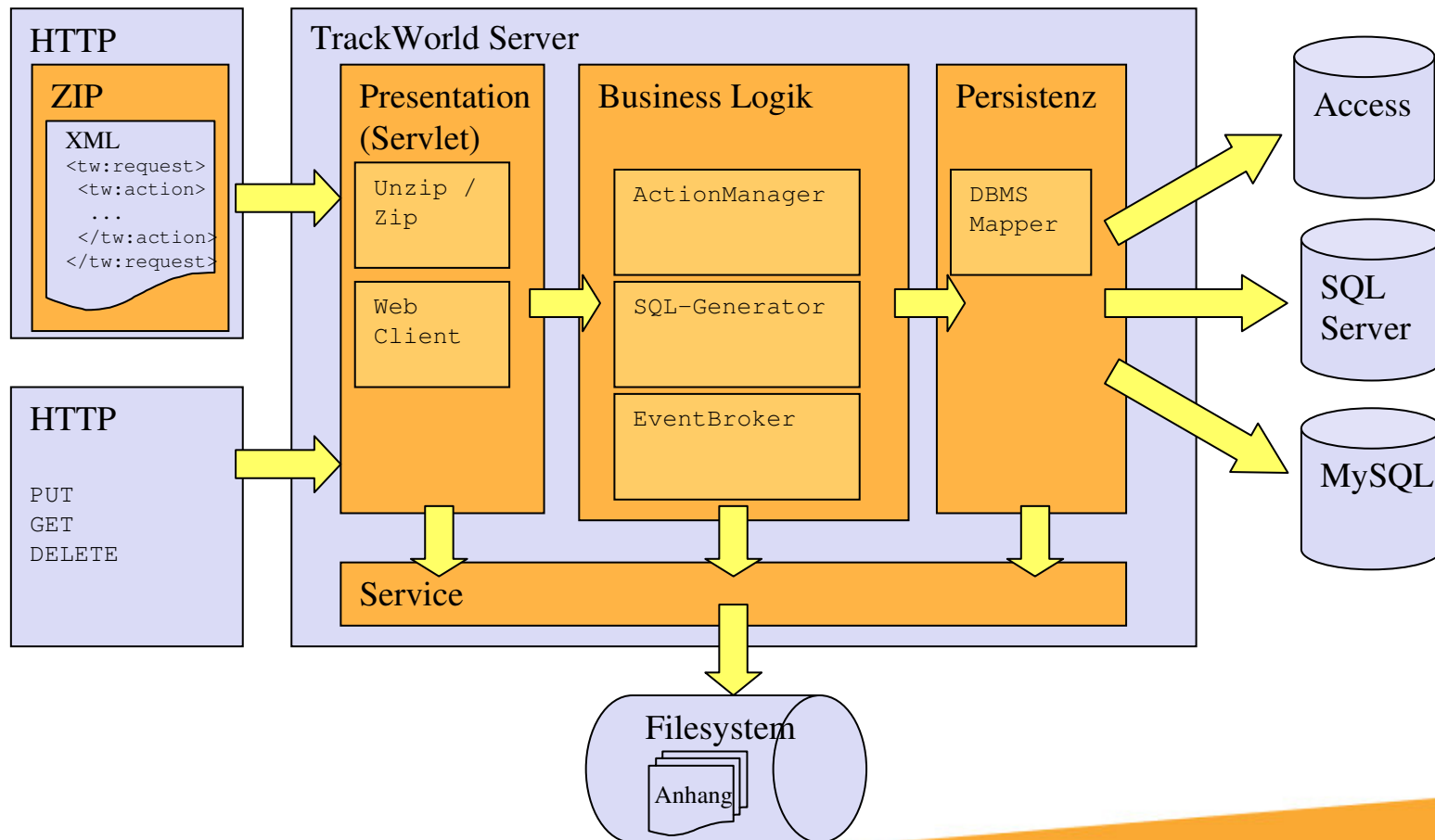
- **Aber kennen wir nicht auch Folgendes?**
 - **Anzahl gemeldeter Fehler durch den Kunden steigt**
 - **Nennenswerte Tests sind kaum vorhanden**
 - **Automatisierte Tests existieren noch weniger**

Ausgangssituation

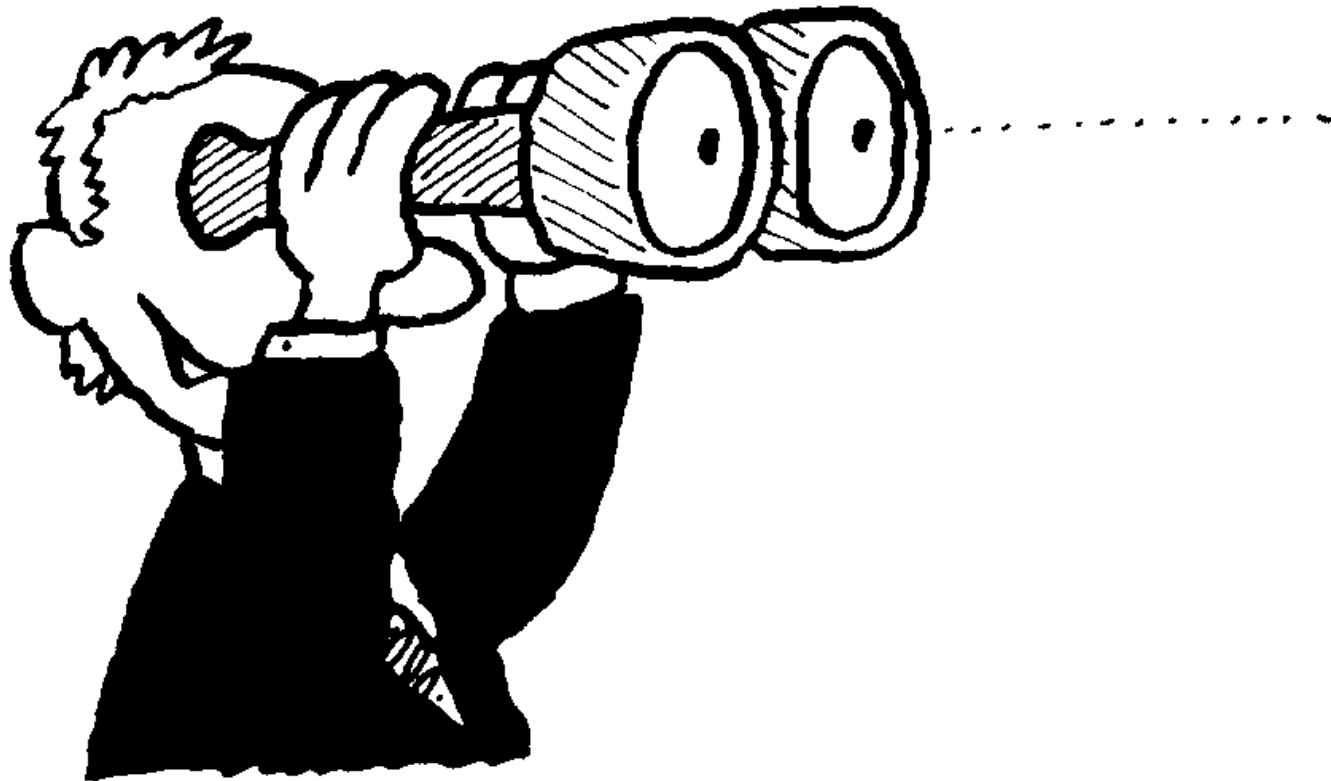
- **Seien Sie unser Reisebegleiter ...**
 - **Betrachtung: Server einer BugTracking-Anwendung**
 - **Löschen aller Tests, die bis dato existieren**
- **Zielsetzung**
 - **Aufbau Testumgebung mit Open-Source von 0 auf 100**
 - **Kontinuierliche, automatisierte**
 - **Integrationstests (externe Interfaces)**
 - **Statische Überprüfungen der Code-Qualität**
 - **Last- und Performancetests**
 - **Unit Tests**
 - **Komponententests (intere Interfaces)**
 - **Implementierungstests**

Ausgangssituation

- **Überblick der Anwendungsarchitektur**



Verwendete Testumgebung



Verwendete Testumgebung

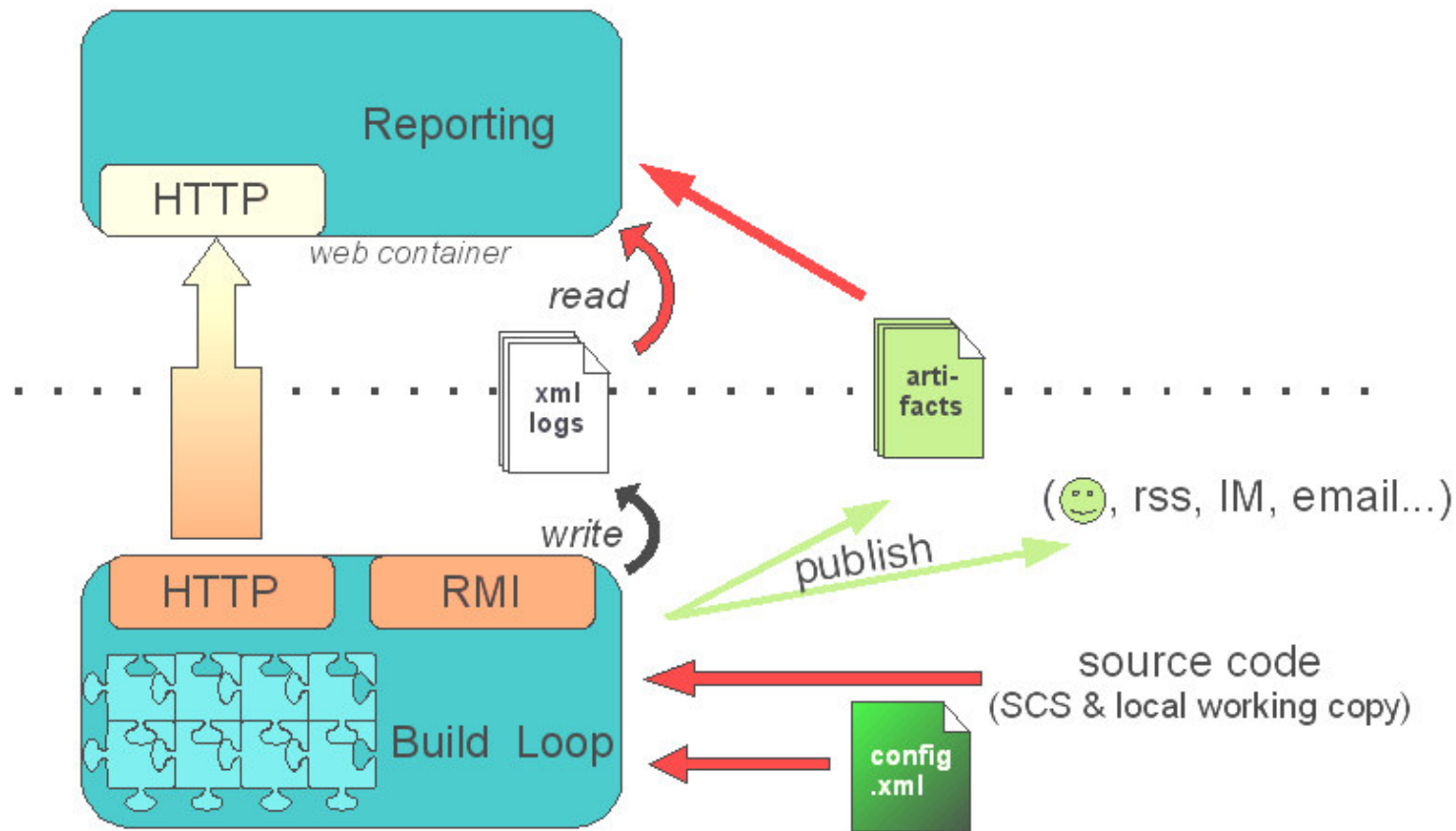
- **In den Hauptrollen ...**
 - **CruiseControl für kontinuierliche Integration**
 - **JMeter für Integrations- und Lasttests**
 - **PMD für statische Überprüfung der Codequalität**
 - **Emma zur Code Coverage Messung**
 - **JUnit darf natürlich nicht fehlen**

Verwendete Testumgebung

- **CruiseControl für kontinuierliche Integration und Tests**
 - **Ermöglicht die Konfiguration von**
 - **Build-Zyklen**
 - **Build-Bedingungen**
 - **Build-Zielen**
 - **Automatisiert folgende Aufgaben**
 - **Build einer Applikation**
 - **Deployment einer Applikation**
 - **Test einer Applikation**
 - **Bereitstellung von Release-Ergebnissen und Reports**

Verwendete Testumgebung

- **CruiseControl für kontinuierliche Integration und Tests**



Verwendete Testumgebung

- **CruiseControl für kontinuierliche Integration und Tests**



cruisecontrol
continuous integration toolkit

Project
commons-math

waiting for next time to build since
12/04/2005 20:40:43

Latest Build
12/04/2005 11:50:32 (build.1)
05/16/2005 06:46:59 (build.0)

RSS

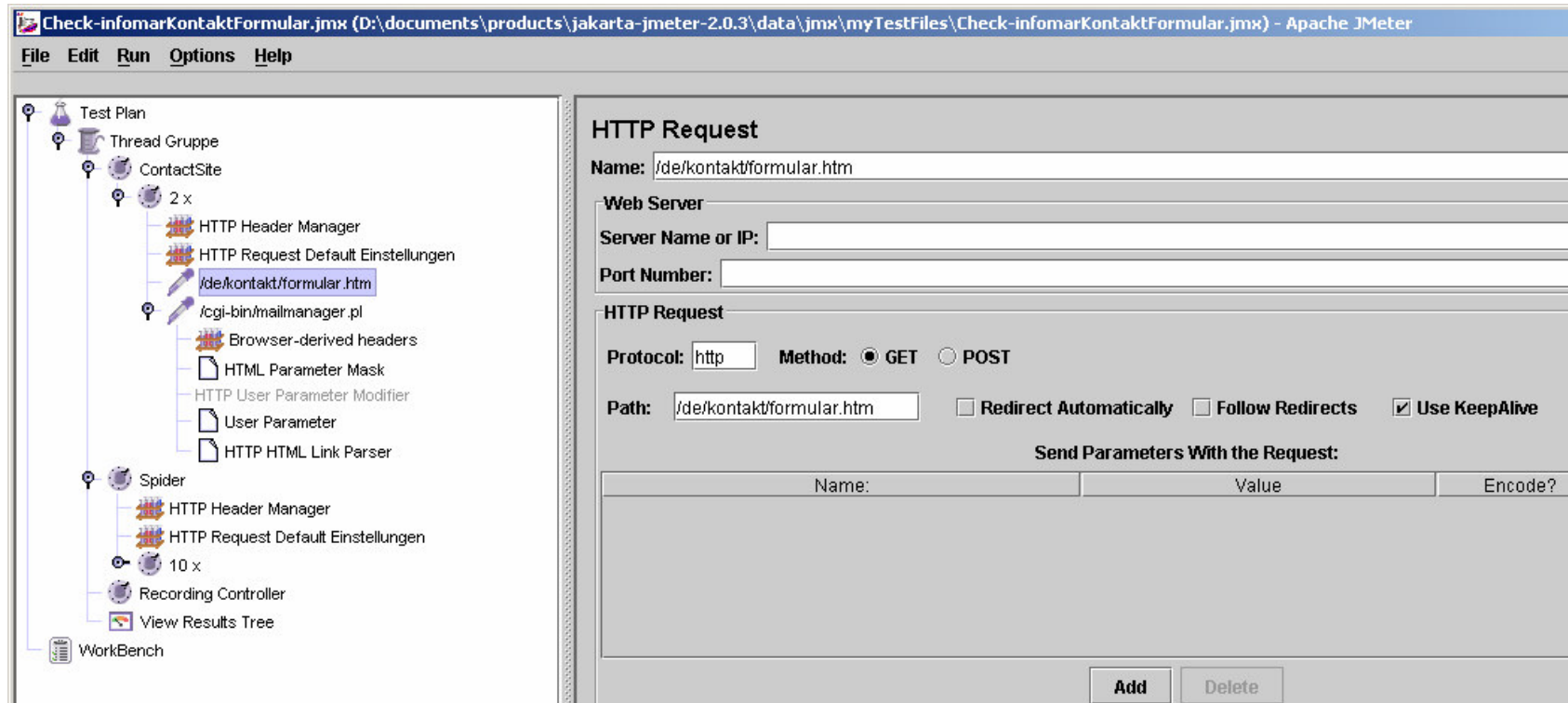
Build Results	Test Results	XML Log File	Metrics	Control Panel
BUILD COMPLETE - build.1				
Date of build:		12/04/2005 11:50:32		
Time to build:		2 minutes 4 second		
Build Artifacts				
Errors/Warnings: (15)				
Note: Some input files use unchecked or unsafe operations. Note: Recompile with -Xlint:unchecked for details.				
D:\Programme\cruisecontrol\2.3.1\projects\jakarta-commons\math\src\test\org\apache\commons\math\stat\CertifiedDataT cast to java.lang.Class for a varargs call cast to java.lang.Class[] for a non-varargs call and to suppress this warning u.getClass().getDeclaredMethod("clear", null).invoke(u, null); ^				
D:\Programme\cruisecontrol\2.3.1\projects\jakarta-commons\math\src\test\org\apache\commons\math\stat\CertifiedDataT cast to java.lang.Object for a varargs call cast to java.lang.Object[] for a non-varargs call and to suppress this warning u.getClass().getDeclaredMethod("clear", null).invoke(u, null); ^				
Note: Some input files use unchecked or unsafe operations. Note: Recompile with -Xlint:unchecked for details. 2 warnings				

Verwendete Testumgebung

- **JMeter für Integrations- und Lasttests**
 - **Viele Treiber: HTTP, SOAP, LDAP, JDBC, TCP, ...**
 - **Ermöglicht verteilte Ausführung auf verschiedenen Rechnern**
 - **Grafisches, wenn auch leicht antiquiertes Frontend**
 - **Einfache Erweiterbarkeit durch Plugins**
 - **Reports anpassbar mit XSLT**
 - **ANT Task**

Verwendete Testumgebung

- **JMeter für Integrations- und Lasttests**



The screenshot shows the Apache JMeter GUI. The left pane displays a Test Plan tree with the following structure:

- Test Plan
 - Thread Gruppe
 - ContactSite
 - 2 x
 - HTTP Header Manager
 - HTTP Request Default Einstellungen
 - /de/kontakt/formular.htm (selected)
 - /cgi-bin/mailmanager.pl
 - Browser-derived headers
 - HTML Parameter Mask
 - HTTP User Parameter Modifier
 - User Parameter
 - HTTP HTML Link Parser
 - Spider
 - HTTP Header Manager
 - HTTP Request Default Einstellungen
 - 10 x
 - Recording Controller
 - View Results Tree

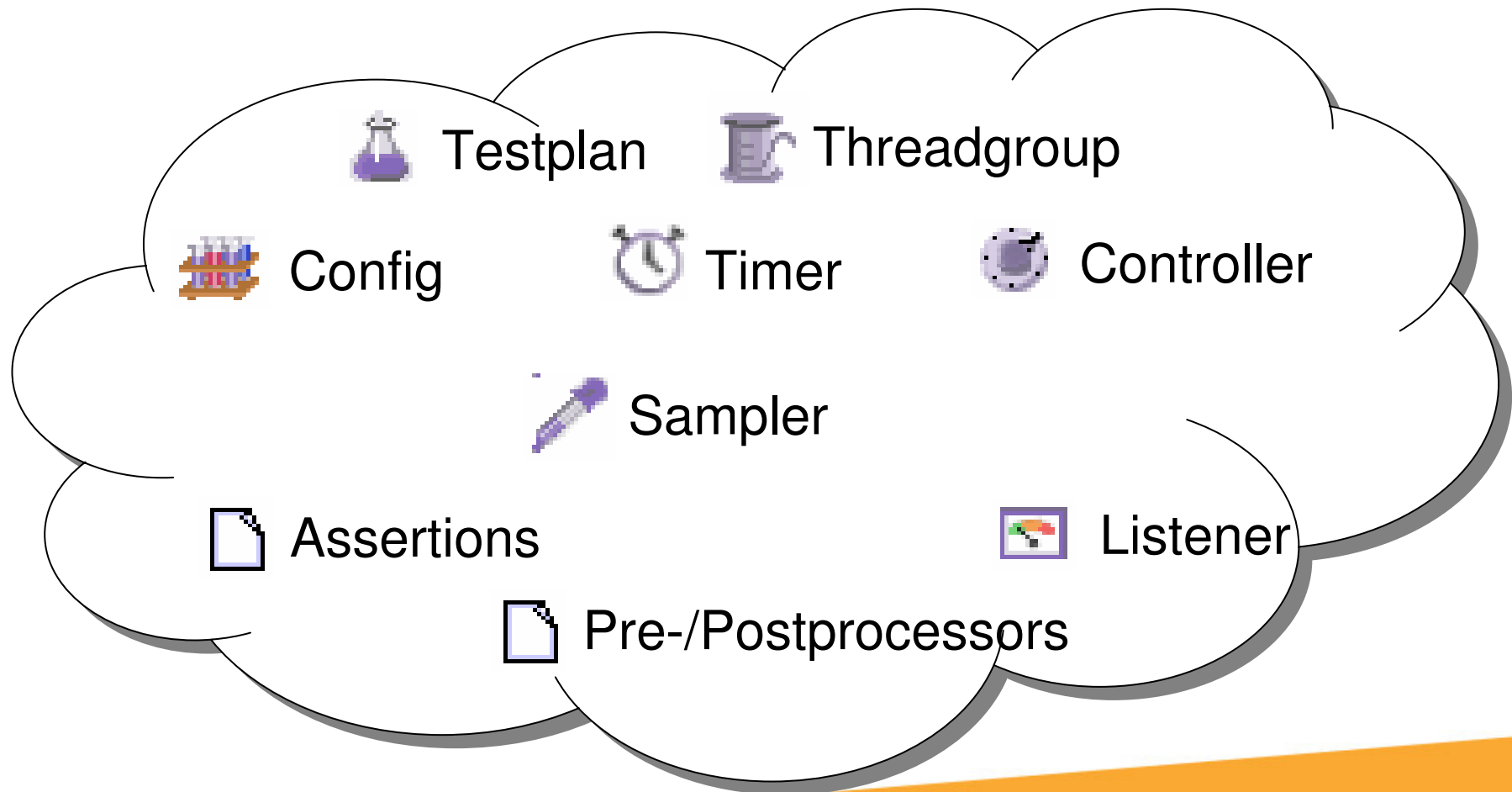
The right pane shows the configuration for the selected **HTTP Request**:

- Name:** /de/kontakt/formular.htm
- Web Server**
 - Server Name or IP: []
 - Port Number: []
- HTTP Request**
 - Protocol: http
 - Method: GET POST
 - Path: /de/kontakt/formular.htm
 - Redirect Automatically
 - Follow Redirects
 - Use KeepAlive
- Send Parameters With the Request:**

Name:	Value	Encode?

Verwendete Testumgebung

- **JMeter für Integrations- und Lasttests**

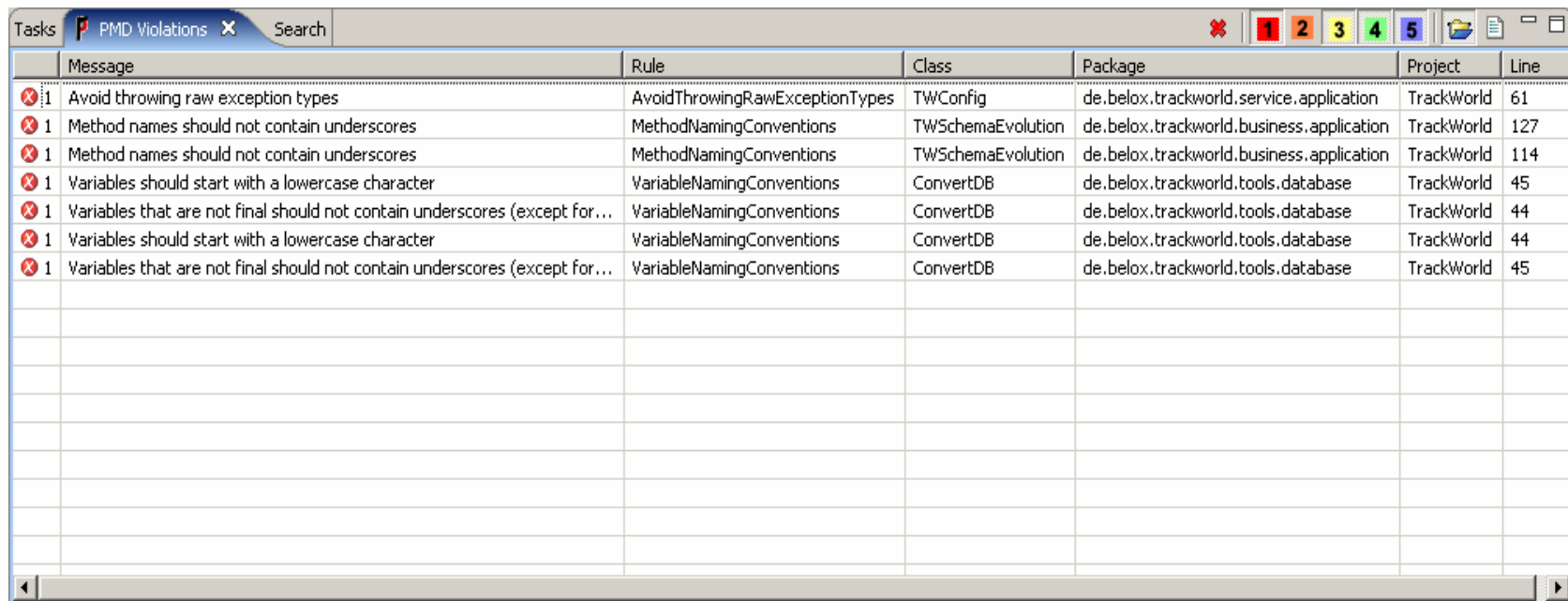


Verwendete Testumgebung

- **PMD für statische Überprüfung der Codequalität**
 - **Findet**
 - **potentielle Laufzeitfehler**
 - **toten Code**
 - **Suboptimalen Code**
 - **Duplizierten Code (Copy Paste Detector)**
 - **PlugIns für verschiedene IDEs (z.B. Eclipse)**
 - **Unterteilung der Regeln in Rulesets**
 - **Eigene Regeln in Java oder XPath realisierbar**
 - **Reports anpassbar mit XSLT**
 - **ANT Task**

Verwendete Testumgebung

- **PMD für statische Überprüfung der Code Qualität**



	Message	Rule	Class	Package	Project	Line
✘ 1	Avoid throwing raw exception types	AvoidThrowingRawExceptionTypes	TWConfig	de.belox.trackworld.service.application	TrackWorld	61
✘ 1	Method names should not contain underscores	MethodNamingConventions	TWSchemaEvolution	de.belox.trackworld.business.application	TrackWorld	127
✘ 1	Method names should not contain underscores	MethodNamingConventions	TWSchemaEvolution	de.belox.trackworld.business.application	TrackWorld	114
✘ 1	Variables should start with a lowercase character	VariableNamingConventions	ConvertDB	de.belox.trackworld.tools.database	TrackWorld	45
✘ 1	Variables that are not final should not contain underscores (except for ...)	VariableNamingConventions	ConvertDB	de.belox.trackworld.tools.database	TrackWorld	44
✘ 1	Variables should start with a lowercase character	VariableNamingConventions	ConvertDB	de.belox.trackworld.tools.database	TrackWorld	44
✘ 1	Variables that are not final should not contain underscores (except for ...)	VariableNamingConventions	ConvertDB	de.belox.trackworld.tools.database	TrackWorld	45

Verwendete Testumgebung

- **Emma zur Code Coverage Messung**
 - **Ermittelt folgende Coverage Werte**
 - **Klassen, Methoden, Block, Zeilen**
 - **Bytecode Instrumentierung**
 - **on-the-fly**
 - **offline**
 - **Ausgabeformate: Text, XML, HTML**
 - **Merge von verschiedenen Coverage Sessions**
 - **ANT Task**

Verwendete Testumgebung

- **Emma zur Code Coverage Messung**

```

EMMA Coverage Report (generated Sat Dec 03 21:58:46 CET 2005)
[all classes]

OVERALL COVERAGE SUMMARY



| name        | class, %   | method, %  | block, %      | line, %     |
|-------------|------------|------------|---------------|-------------|
| all classes | 100% (3/3) | 100% (7/7) | 95% (120/126) | 90% (26/29) |



OVERALL STATS SUMMARY

total packages:      2
total executable files: 3
total classes:      3
total methods:      7
total executable lines: 29

COVERAGE BREAKDOWN BY PACKAGE



| name            | class, %   | method, %  | block, %     | line, %      |
|-----------------|------------|------------|--------------|--------------|
| search          | 100% (2/2) | 100% (4/4) | 91% (64/70)  | 83% (15/18)  |
| default package | 100% (1/1) | 100% (3/3) | 100% (56/56) | 100% (11/11) |



[all classes]
EMMA 2.0.5312 (C) Vladimir Roubtsov

```

Verwendete Testumgebung

- **JUnit natürlich**
 - **Mit der Magie des grünen Balkens**
 - **Zentrale Elemente**
 - **Assert mit** `assertEquals`, `assert (Not) Null`, `assert (Not) Same`
 - `Test`, `TestCase`, `TestSuite`
 - **TestRunner**
 - **ANT Task**
 - **Version 4 Features**
 - **Annotations statt Ableitung und Namenskonvention**
 - `@Test`, `@Before(Class)`, `@After(Class)`, `@Ignore`
 - **Timeouts und expected exceptions als Parameter**
 - **Eigene Runner fallen weg**

Verwendete Testumgebung

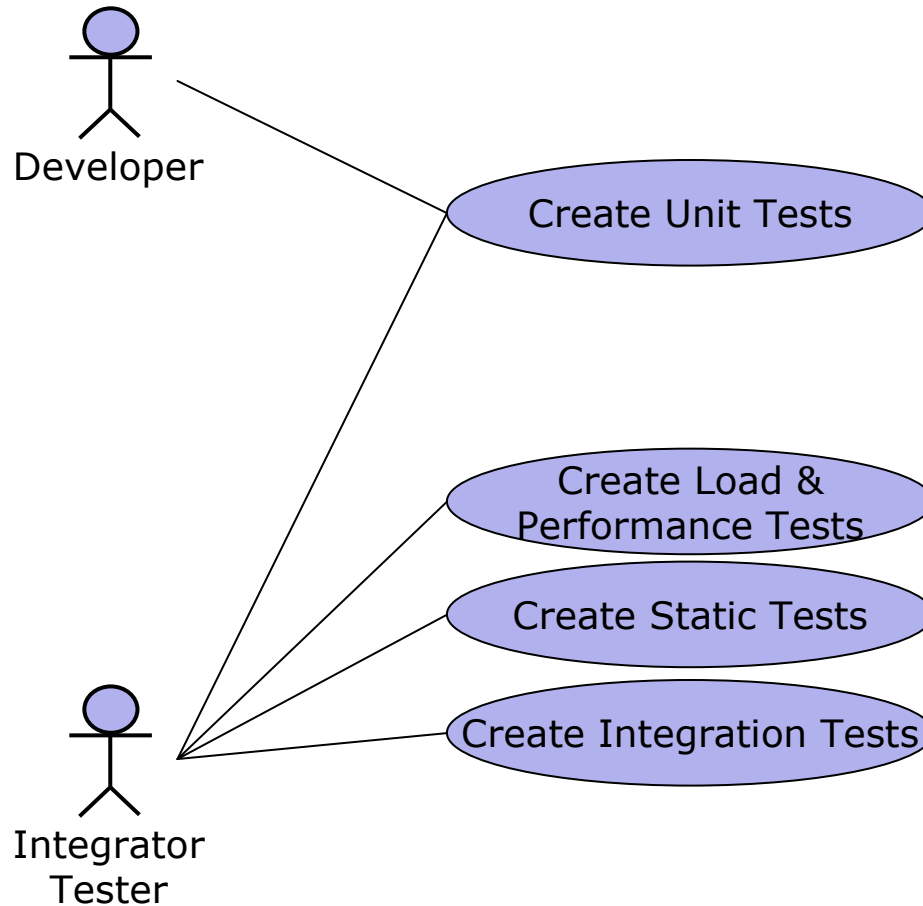
- **In den Nebenrollen**
 - **JavaNCSS**
 - **Easymock und Mockobjects**
 - **HSQLDB InMemory Datenbank**
- **Auf der Ersatzbank**
 - **Fit & FitNesse**
 - **Wiki**

Tests und Testautomatisierung



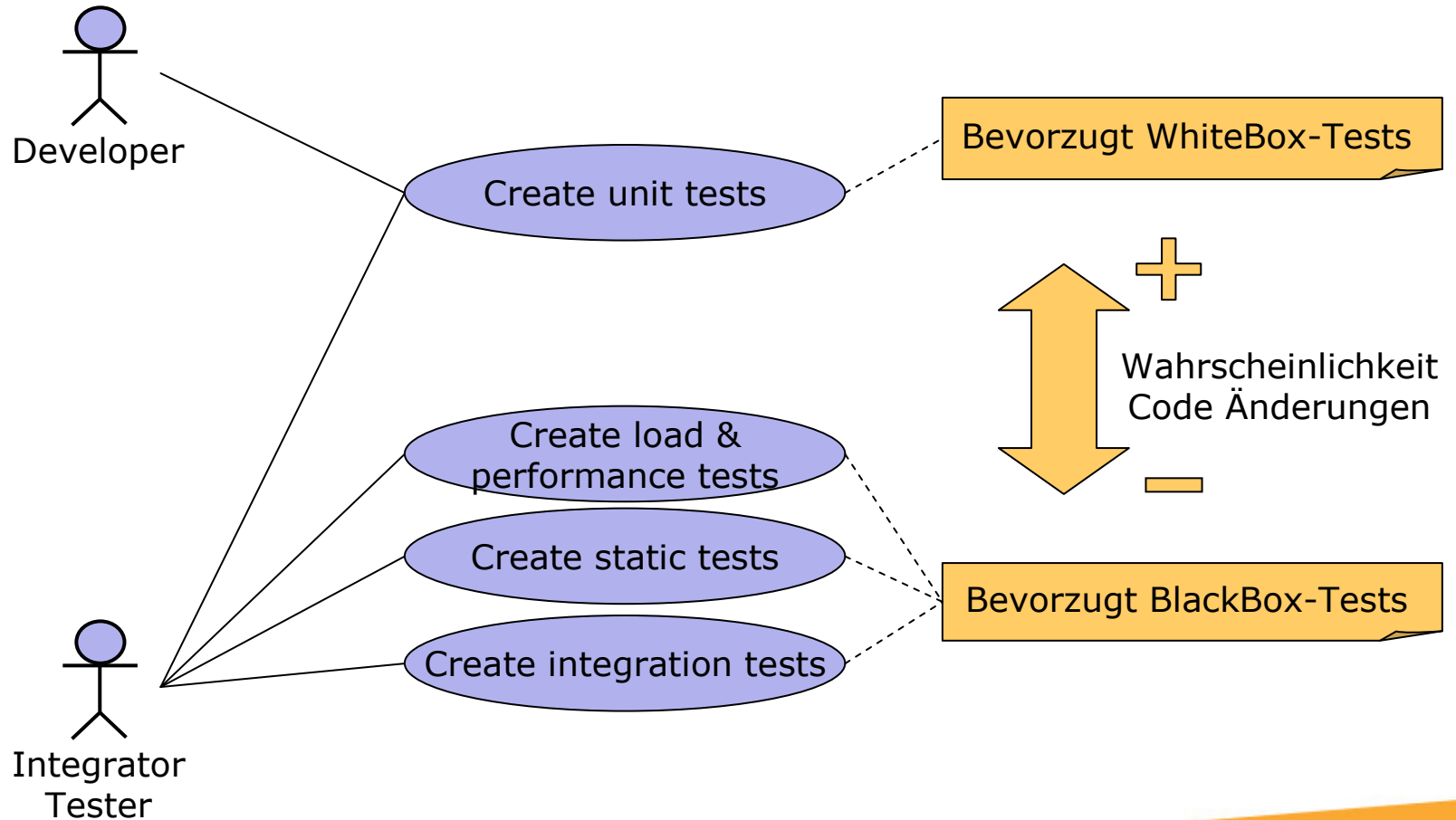
Tests und Testautomatisierung

- **Übersicht Testerzeugung**



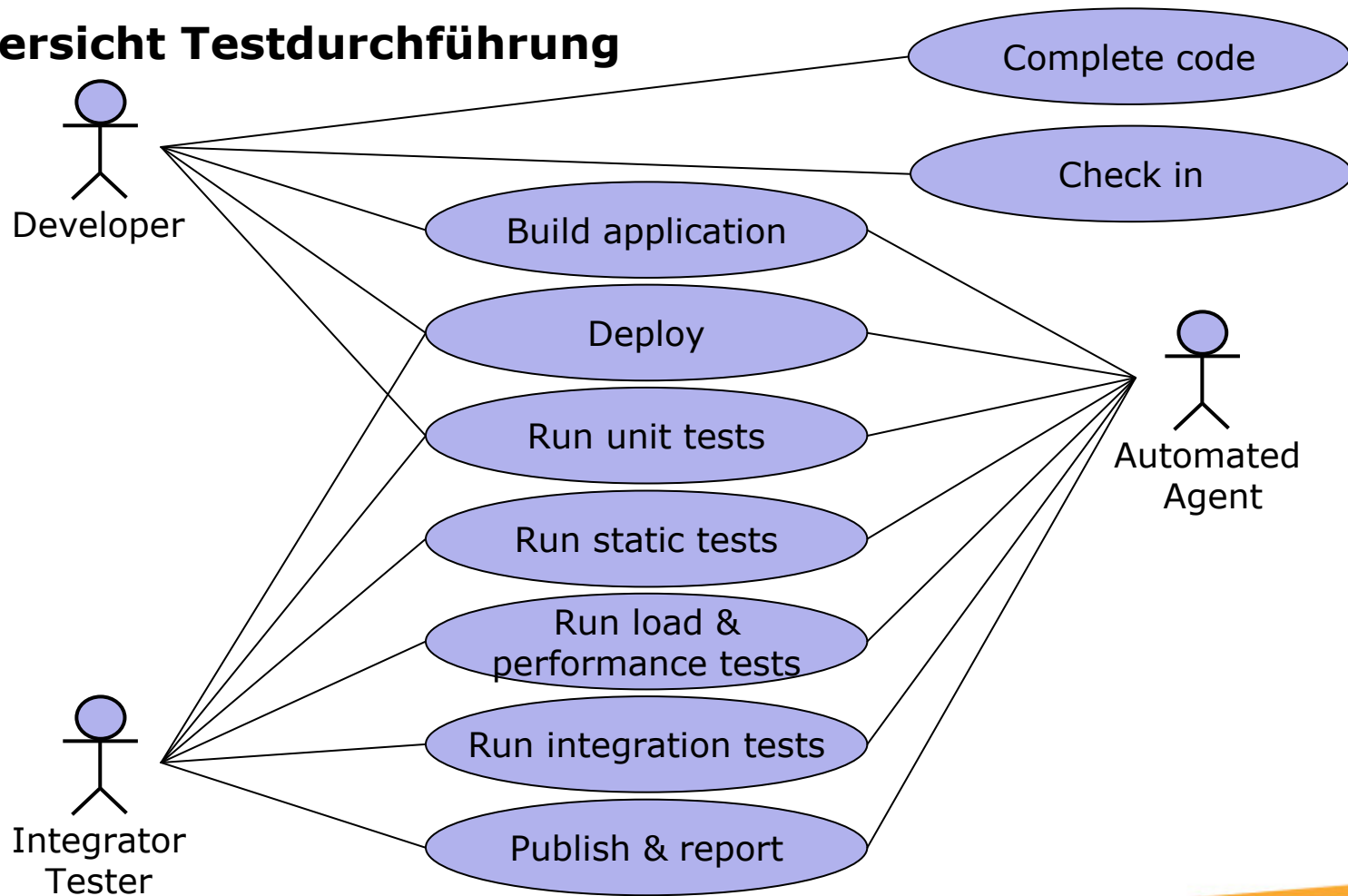
Tests und Testautomatisierung

- **Testerzeugung und Wahrscheinlichkeit für Code Änderungen**



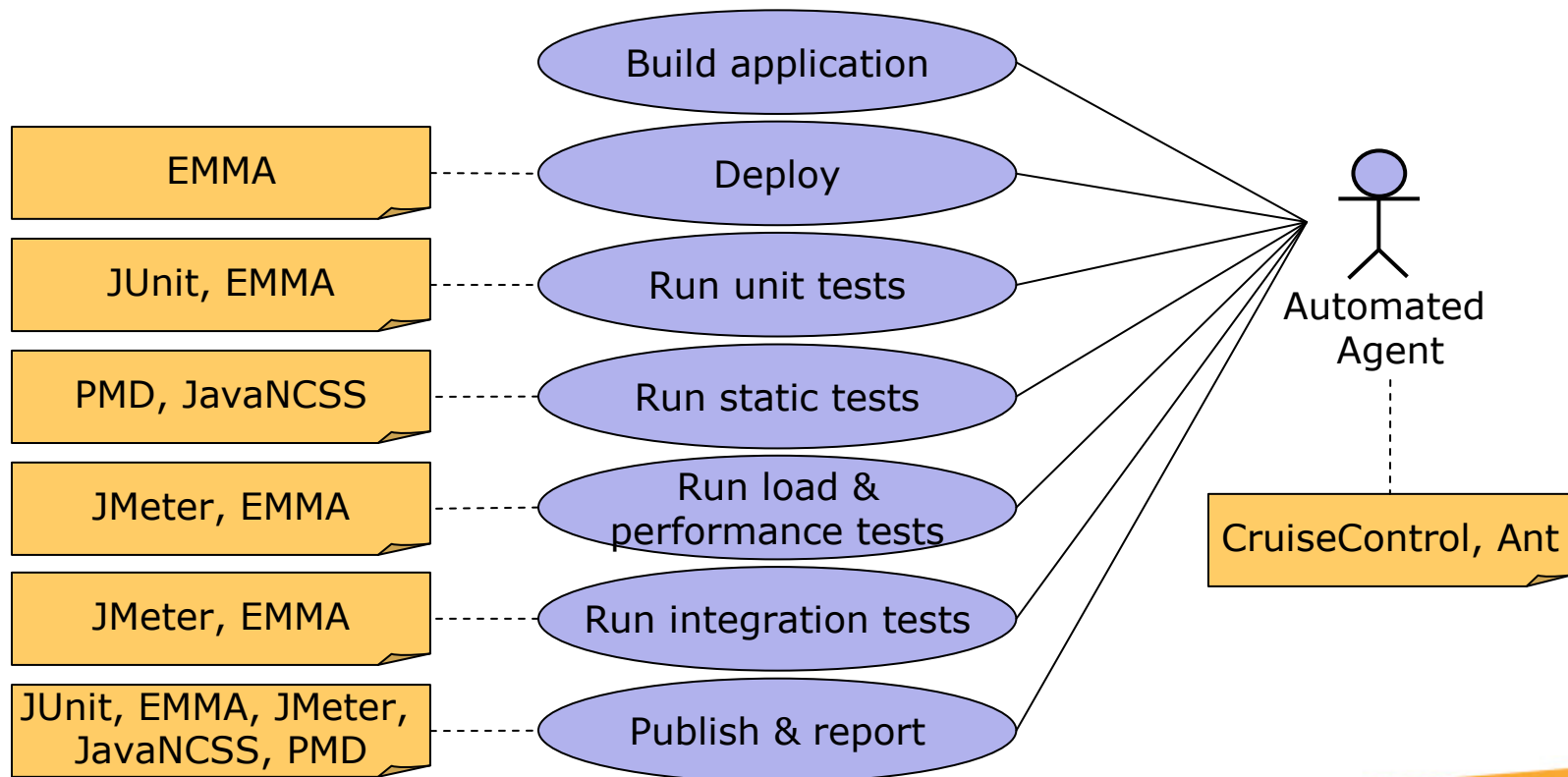
Tests und Testautomatisierung

- **Übersicht Testdurchführung**



Tests und Testautomatisierung

- Testdurchführung und verwendete Werkzeuge**



Tests und Testautomatisierung

- **Integrationstests (externe Schnittstellen)**
 - **Aufgaben**
 - **Dokumentation der Server-Schnittstelle**
 - **Entwicklung von JMeter-Plugin**
 - **Speicherung von User-Aktionen als JMeter Testpläne**
 - **Verwendung des JMeter ANT Tasks**
 - **Automatisierung der Tests mittels CruiseControl**
 - **Nutzen**
 - **Nicht-invasiver Test, der Server-Code unverändert läßt**
 - **Gute Abdeckung der vom Client benötigten Funktionalität**
 - **Sicherere Ausgangsbasis für weitere Änderungen**

Tests und Testautomatisierung

- **Statische Überprüfung der Code-Qualität**
 - **Aufgaben**
 - **Einrichtung von PMD, JNCSS & EMMA in CruiseControl**
 - **Einschränkung von PMD auf wichtige Probleme**
 - **Instrumentierung des Produktiv-Codes für EMMA**
 - **Verwendung der ANT-Tasks für jedes Tool**
 - **Automatisierung der Tool-Aufrufe in Cruise-Control**
 - **Nutzen**
 - **Unterstützung zur Suche von**
 - **Fehlern, Code Smells, Copy-Paste Code**
 - **Überwachung von Coding Guide Lines**
 - **Beurteilung der Testabdeckung**

Tests und Testautomatisierung

- **Last- und Performancetests**
 - **Aufgaben**
 - **Erstellung von Lastprofilen**
 - **Typen und Anzahl von Anwendern**
 - **Aktionen pro Anwendertyp**
 - **Verteilung der Anwendertypen über Zeiträume**
 - **Lasttests aus JMeter Testplänen für GUI-Aktionen**
 - **Performancetests für Einzel- und Verbund-Aktionen**
 - **Einbindung in CruiseControl analog Integrationstest**
 - **Nutzen**
 - **Aussagen zur Belastbarkeit der Anwendung**
 - **Performance-Werte für Server-Schnittstelle**

Tests und Testautomatisierung

- **Unit Tests (interne Schnittstellen)**
 - **Aufgaben**
 - **JUnit zum Test einzelner Architekturschichten**
 - **EasyMock, MockObjects zur Simulation genutzter Schichten**
 - **Einbindung inMemory-Datenbank**
 - **Anpassung von Code zur besseren Testbarkeit**
 - **Nutzen**
 - **Detailgenauere Testabdeckung**
 - **Bessere Lokalisierbarkeit von Fehlern**

Und so läuft's...

- **Es folgen Beispiele, wie die Tools**
 - **In Ant eingebunden werden**
 - **Mit CruiseControl automatisiert werden**



Java NCSS Ant-Task

```

<taskdef name="javancss" classname="javancss.JavancssAntTask"/>

<target name="run-jncss">
  <property name="oop.programs.dir" value="C:/projects/oop2006/programs" />

  <javancss
    srcdir                = "src"
    generateReport        = "true"
    abortOnFail           = "false"
    outputfile            = "javancss_metrics.xml"
    format                = "xml"
    includes              = "**/*.java"
    excludes              = "**/*Test.java" />

  <xslt in = "javancss_metrics.xml"
    out = "javancss_metrics.html"
    style = "${oop.programs.dir}\javancss21.41\xslt\javancss2html.xsl" />
</target>

```

PMD Ant-Task

```

<taskdef name="pmd"    classname="net.sourceforge.pmd.ant.PMDTask"/>

<target name="run-pmd">

  <property name="oop.programs.dir" value="C:/projects/oop2006/programs" />

  <pmd>
    <ruleset>basic</ruleset>
    <formatter type      = "xml"
              toFile    = "pmd-report.xml"/>
    <fileset dir = "src">
      <include name = "**/*.java"/>
    </fileset>
  </pmd>

  <xslt  in = "pmd-report.xml"
        out = "pmd-report.html" />
  <xslt  style = "${oop.programs.dir}\pmd-3.4\etc\xslt\pmd-report-trackworld.xslt" />
</target>

```

Weitere Regeln z.B.
 strictexception
 design
 unusedcode
 finalizers
 sunsecure

Emma Ant-Task – Instrumentierung

```
<taskdef name="emma" resource="emma_ant.properties" />

<target name="run-emma-instrument">

  <emma enabled="true" >

    <instr      instrpath = "deploy/lib/belox-trackworld.jar"
                outfile   = "deploy/lib/coverage.em"
                mode      = "overwrite">

      <filter excludes = "*Test *.test.*" />
    </instr>

  </emma>
</target>
```

Emma Ant-Task – Reporting

```

<target name="run-emma-report">
  <emma enabled="true" >
    <report      sourcepath      = "src"
                 sort            = "+block,+name,+method,+class,+line"
                 metrics         = "method:70,block:80,line:80,class:100">

      <infileset dir="deploy" includes="*.em, *.ec" />
      ↓
      <html      outfile      = "coverage.html"
                 depth       = "method"
                 columns     = "name,class,method,block,line" />

    </report>
  </emma>
</target>

```

Weitere mögl. Elemente:
txt
xml

JMeter Ant-Task

```
<taskdef name="jmeter" classname="org.programmerplanet.ant.taskdefs.jmeter.JMeterTask"/>

<target name="run-jmeter">

  <property name="oop.programs.dir" value="C:/projects/oop2006/programs" />

  <jmeter jmeterhome      = "${oop.programs.dir}/jakarta-jmeter-2.1.1"
        resultlog        = "JMeterResults.jtl">

    <property name = "twServerIp"   value = "localhost"/>
    <property name = "twServerPort" value = "80"/>

    <testplans dir = "jmeter"      includes = "*.jmx"/>
  </jmeter>

  <xslt in = "JMeterResults.jtl"
        out = "JMeterResults.html" />
  style = "${oop.programs.dir}/jakarta-jmeter-2.1.1/extras/jmeter-results-detail-report.xsl" />

</target>
```

CruiseControl Konfiguration

```
<cruisecontrol>
  <property name="cchome"
    value="..\..\programs\cruisecontrol-2.3.1" />

  <project name      = "TrackWorld"
    buildafterfailed = "true">

    <modificationset quietperiod      = "5"
      requiremodification = "true">
      <cvs localWorkingCopy="projects/TrackWorld"/>
    </modificationset>

    <schedule interval = "30">
      <ant anthome = "${cchome}\apache-ant-1.6.3"
        buildfile   = "build.xml"
        target      = "deploy test"
        uselogger   = "true"
        usedebug    = "false" />
    </schedule>
  </project>
</cruisecontrol>
```

```
  <log>
    <merge dir="unittest"/>
  </log>

  <publishers>
    <artifactspublisher
      dest      = "artifacts/TrackWorld"
      file      = "JMeterResults.html" />
  </publishers>

</project>

</cruisecontrol>
```

Fazit / Zusammenfassung



Fazit / Zusammenfassung

- + Schneller Aufbau einer Testumgebung**
Ohne Installationsaufwand (Copy-Deployment)
- + Automatisierbarkeit**
ANT / CruiseControl
Kontinuierlicher Build, Test, Review
- + Offene Formate für Reporting**
XSLT-Templates bereits vorhanden
Weitere sind im Web verfügbar
- + Unterstützung nahezu aller Source Control Systems**
- + Permanentes Review**

- Kleinere Bugs**
Tiefe Verzeichnisstruktur verwirrt CruiseControl
XSLT-Stylesheets nicht immer sauber programmiert

Weiterführende Links und Quellen

- **Bücher**

- Testgetriebene Entwicklung mit JUnit & FIT, Frank Westphal, dpunkt Verlag 2005
- Softwaretests mit JUnit, Johannes Link, dpunkt Verlag 2005
- J2EE-Entwicklung mit Open-Source-Tools, Martin Backschat / Stefan Edlich, Spektrum Verlag, 2004 (ein paar Seiten)

- **Links**

- CruiseControl <http://cruisecontrol.sourceforge.net>
- JMeter <http://jakarta.apache.org/jmeter>
- EMMA <http://emma.sourceforge.net>
- PMD <http://pmd.sourceforge.net>
- JUnit <http://www.junit.org>
- JavaNCSS <http://www.kclee.de/clemens/java/javancss>
- FIT <http://fit.c2.com> und FITnesse <http://fitnesse.org>

- **Artikel**

- Paper Rutar et al: „A Comparison of Bug Finding Tools for Java“
<http://www.cs.umd.edu/~jfoster/papers/issre04.pdf>

Fragen?

Aber gern ...



Martin Heider
Infomar software
mh@infomar.de
www.infomar.de



andreas oetjen
belox software gmbh
oetjen@belox.de
www.belox.de

