



Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

ARTEM EGER

BUILD-SYSTEME IN JAVA

MAVEN & ANT

GLIEDERUNG

- (1) Motivation und Überblick
- (2) Bauen mit Ant
- (3) Praxisbeispiel Ant
- (4) Bauen mit Maven
- (5) Praxisbeispiel Maven
- (6) Ausblick



(1) MOTIVATION UND ÜBERBLICK

Was ist ein Build?



(1) MOTIVATION UND ÜBERBLICK

1. Kompilieren:

Übersetzung von Quelltext in Maschinencode

(1) MOTIVATION UND ÜBERBLICK

1. Kompilieren

2. Linking:

Verknüpfen von Maschinencode und Bibliotheken zu einer ausführbaren Datei



(1) MOTIVATION UND ÜBERBLICK

1. Kompilieren

2. Linking

3. Building:

Kompilieren + Linking + etc.



(1) MOTIVATION UND ÜBERBLICK

Welche Schritte gehören zu einem Build?

(1) MOTIVATION UND ÜBERBLICK

- Generierung von Quelltext
- Kompilieren des Quelltextes
- Kompilieren der Tests
- Ausführen der Tests
- Packaging
- Generierung der Dokumentation

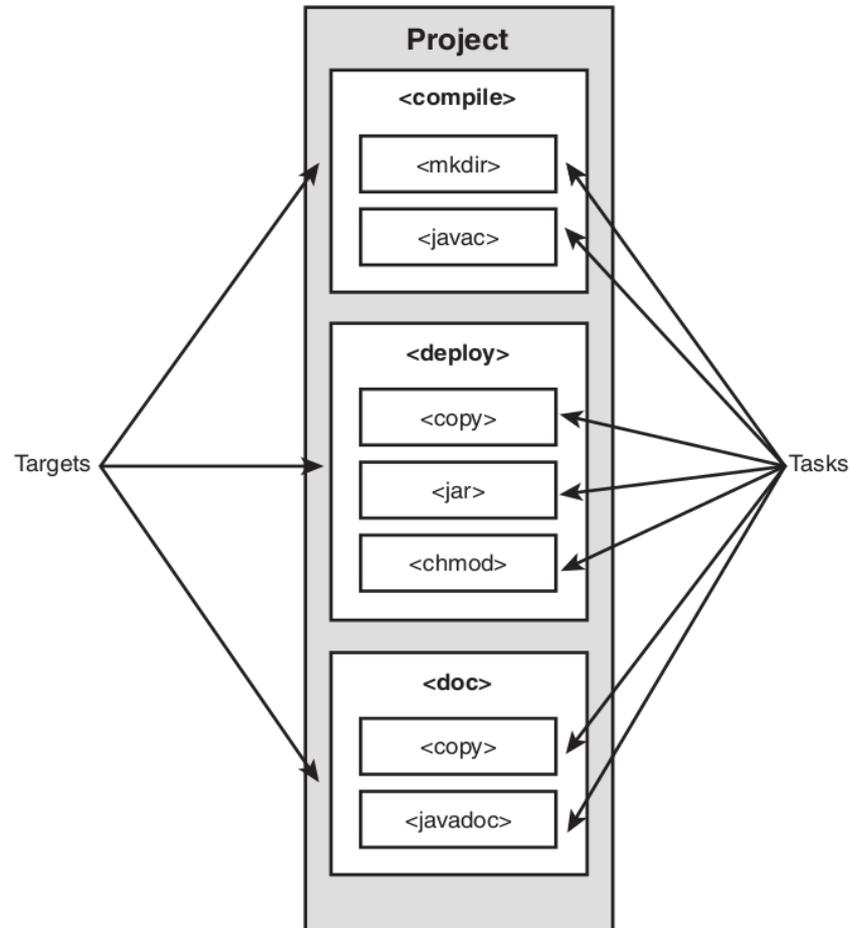


(1) MOTIVATION UND ÜBERBLICK

Wofür brauchen wir Buildsysteme?

(2) BAUEN MIT ANT

- Ant ist Java zentriert
- Definiert wird in XML im Buildfile
- Projekt besteht aus einem oder mehreren Targets
- Targets bestehen aus einem oder mehreren Tasks



(2) BAUEN MIT ANT

Ant Projekte erfordern eine
„build.xml“

```
<?xml version="1.0" encoding="UTF-8">
```

Jede „build.xml“ beginnt mit der
Definition der XML-Version für den
Parser

(2) BAUEN MIT ANT

Anschließend muss das Projekt angelegt werden

```
<?xml version="1.0" encoding="UTF-8">  
<project name="TestProjekt" default="test.ant" basedir=".">  
  
</project>
```

Ein Projekt kann drei Attribute haben:

name default basedir

(2) BAUEN MIT ANT

Hiernach können Targets angelegt werden

Targets können folgende Attribute haben

name depends if unless description

```
<?xml version="1.0" encoding="UTF-8">  
<project name="TestProjekt" default="test.ant" basedir=".">  
    <target name="test.ant" description="A simple build file to test ant.">  
    </target>  
</project>
```

(2) BAUEN MIT ANT

Im Target können Tasks definiert werden

Es wird unterschieden zwischen Core Tasks und Optional Tasks

```
<?xml version="1.0" encoding="UTF-8">  
<project name="TestProjekt" default="test.ant" basedir=".">  
  <target name="test.ant" description="A simple build file to test ant.">  
    <echo message="Ant is working properly" />  
  </target>  
</project>
```

(2) BAUEN MIT ANT

- Wenn keine Defaultparameter benutzt wurden kann der Buildprozess folgendermaßen gestartet werden:

ant -buildfile mybuildfile.xml test.ant

- Falls alles Default benannt ist reicht ein einfaches:

ant



(3) PRAXISBEISPIEL ANT

(4) BAUEN MIT MAVEN

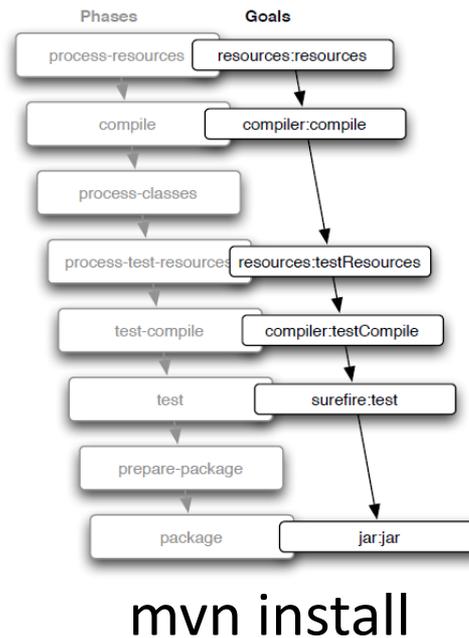
Kernkonzepte – Plugins und Goals



`mvn archetype:generate`

(4) BAUEN MIT MAVEN

Kernkonzepte – Maven Default Lifecycle



(4) BAUEN MIT MAVEN

Kernkonzepte – Lifecycle Phases



(4) BAUEN MIT MAVEN

Kernkonzepte – Coordinates

```
<groupId>mavenbook</groupId>  
<artifactId>my-app</artifactId>  
<packaging>jar</packaging>  
<version>1.0-SNAPSHOT</version>
```



(4) BAUEN MIT MAVEN

Kernkonzepte – Repository and Dependency

<https://mvnrepository.com>

(4) BAUEN MIT MAVEN

Der erste Schritt ist einen Archetype auszuwählen

Dann wird dafür automatisch der entsprechende Source Code und die passende Struktur generiert

Entweder einfach grafisch mit Eclipse

Oder....

Archetype ArtifactIds

maven-archetype-archetype

maven-archetype-j2ee-simple

maven-archetype-plugin

maven-archetype-plugin-site

maven-archetype-portlet

maven-archetype-quickstart

maven-archetype-simple

maven-archetype-site

maven-archetype-site-simple

maven-archetype-webapp

Description

An archetype to generate a sample archetype.

An archetype to generate a simplified sample J2EE application.

An archetype to generate a sample Maven plugin.

An archetype to generate a sample Maven plugin site.

An archetype to generate a sample JSR-268 Portlet.

An archetype to generate a sample Maven project.

An archetype to generate a simple Maven project.

An archetype to generate a sample Maven site which demonstrates some of the supported document types like APT, XDoc, and FML and demonstrates how to i18n your site.

An archetype to generate a sample Maven site.

An archetype to generate a sample Maven Webapp project.

(4) BAUEN MIT MAVEN

mvn archetype:generate

-DgroupId=Testprojekt

-DartifactId=Test

-Dpackage=org.test.testprojekt

-Dversion=1.0

-DarchetypeArtifactId=maven-archetype-quickstart



(4) BAUEN MIT MAVEN

```
<project xmlns="http://maven.apache.org/POM/4.0.0"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0  
http://maven.apache.org/xsd/maven-4.0.0.xsd">  
  
<modelVersion>4.0.0</modelVersion>  
  
  
  
</project>
```



(4) BAUEN MIT MAVEN

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

  <groupId>Testprojekt</groupId>

  <artifactId>Test</artifactId>

  <version>1.0</version>

  <packaging>jar</packaging>

  <name>Test</name>

  <url>http://maven.apache.org</url>

</project>
```



(4) BAUEN MIT MAVEN

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

  <dependencies>

    <dependency>

      <groupId>junit</groupId>

      <artifactId>junit</artifactId>

      <version>3.8.1</version>

      <scope>test</scope>

    </dependency>

  </dependencies>

</project>
```



(5) PRAXISBEISPIEL MAVEN

(6) AUSBLICK



- Release 2012
- Kombiniert die Flexibilität von Ant mit dem Komfort von Maven
- Benutzt kein XML



QUELLEN

<https://hbtsolutionz.files.wordpress.com/2014/01/16960-code-1920x1200-computer-wallpaper.jpg>

Zugriff am 24.10.2016

<https://www.heise.de/developer/artikel/Continuous-Integration-mit-Java-und-JavaScript-Teil-1-Die-Kunst-der-Versionierung-2853319.html?artikelseite=2>

Zugriff am 26.10.2016

<http://stackoverflow.com/questions/2310261/what-is-the-difference-between-compile-code-and-executable-code>

Zugriff am 26.10.2016

<http://www.bennyn.de/programmierung/java/beispiel-fur-apache-ant-build-script.html>

Zugriff am 26.10.2016



QUELLEN

<https://technologyconversations.files.wordpress.com/2014/06/gradle.png>

Zugriff am 1.11.2016

<https://technologyconversations.com/2014/06/18/build-tools/>

Zugriff am 1.11.2016

[1] **G. Niemeyer und J. Poteet**, Extreme Programming with AntBuilding and Deploying Java TMAApplications with JSP TM , EJB TM , XSLT, xDoclet, and JUnit, 1. Hrsg., Indianapolis: Sams Publishing, 2003.

[2] **J. Humble und D. Farley**, Continuous delivery : reliable software releases through build, test, and deployment automation, 1. Hrsg., Crawfordsville: Pearson Education, 2010.

[3] **T. O'Brien, J. v. Zyl, B. Fox, J. Casey, J. Xu und T. Locher**, Maven by Example: An Introduction to Apache Maven, 1. Hrsg., Silver Spring: Sonatype, 2010.