

Build Tools

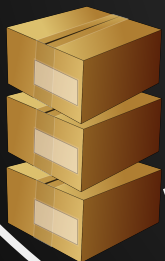
Definition / Aufgaben / Begründung

Produktentwicklungszyklus

Anforderungen

verstehen
bearbeiten
zerlegen

Features / Storys



Entwicklungsiteration (Sprint / Projekt)

programmieren
erstellen
verifizieren



releasen (veröffentlichen)
deploy (installieren)

Was ist ein Build Tool?

“A build tool is a programming utility that is used when building a new version of a program. For example, make is a popular open source build tool that uses makefile, another build tool, to ensure that source files that have been updated (and files that are dependent on them) will be compiled into a new version (build) of a program.”

Quelle: <http://searchsoftwarequality.techtarget.com/definition/build-tool>

Was ist ein Build Tool?

“Build tools are programs that **automate** the **creation** of **executable** applications from **source code**. Building incorporates **compiling**, **linking** and **packaging** the code into a **usable** or **executable** form. In small projects, developers will often manually invoke the build process. This is not practical for larger projects, where it is very hard to keep track of what needs to be built, in what **sequence** and what **dependencies** there are in the building process. Using an **automation** tool allows the build process to be more **consistent**.”

Frei interpretiert:

Ein “Build Tool” ermöglicht formalisierten Beschreibung und wiederholbaren Ausführung des Software-Erstellungsprozesses.

Build Tool - Wozu?

- Das "Works on my machine"-Problem
- Nachvollziehbarkeit
- Reproduzierbarkeit
- Automatisierbarkeit

Ziel: Checkout / Bauen / Testen / Starten /
Bearbeiten

Alternative: viele manuelle wiederholte Arbeiten
um mit dem Projekt zu arbeiten -> Zeitverlust

Was ist kein Build-Tool?

- IDE (Integrated Development Environment) -
> Eclipse / Netbeans / IntelliJ IDEA sind
keine Build-Tools
- Skripte -> eine wilde Sammlung von Skripten
ist/sind kein Build-Tool

Aufgaben

- Kompilieren
- Paketieren
- Abhängigkeitsmanagement
- Verteilungsmanagement
- Testausführung
- Projektmanagement
- Deployment

Kompilieren / Übersetzen

- übersetzen des Quellcodes in Maschinsprache / Byte Code

z. B. javac, groovyc, gcc, etc.

Alternative:

- Manuelle Ausführung des Compilers (javac)
- manuelles Zusammenbauen des Classpath für den Übersetzungsvorgang

Paketieren

Zusammenfügen des kompilierten Codes, der Ressourcen und gegebenenfalls der Bibliotheken zu einem Artefakt

z. B.: EXE, JAR, WAR, EAR, ZIP

Alternative:

- Manuelles "zippen" der Sourcen und erstellen einer MANIFEST-Datei (JAR), etc.
- Manuelles Aktualisieren der Manifest-Datei
- etc.

Abhängigkeitsmanagement

Verwalten von Abhängigkeiten (dependencies) in den jeweiligen Versionen.

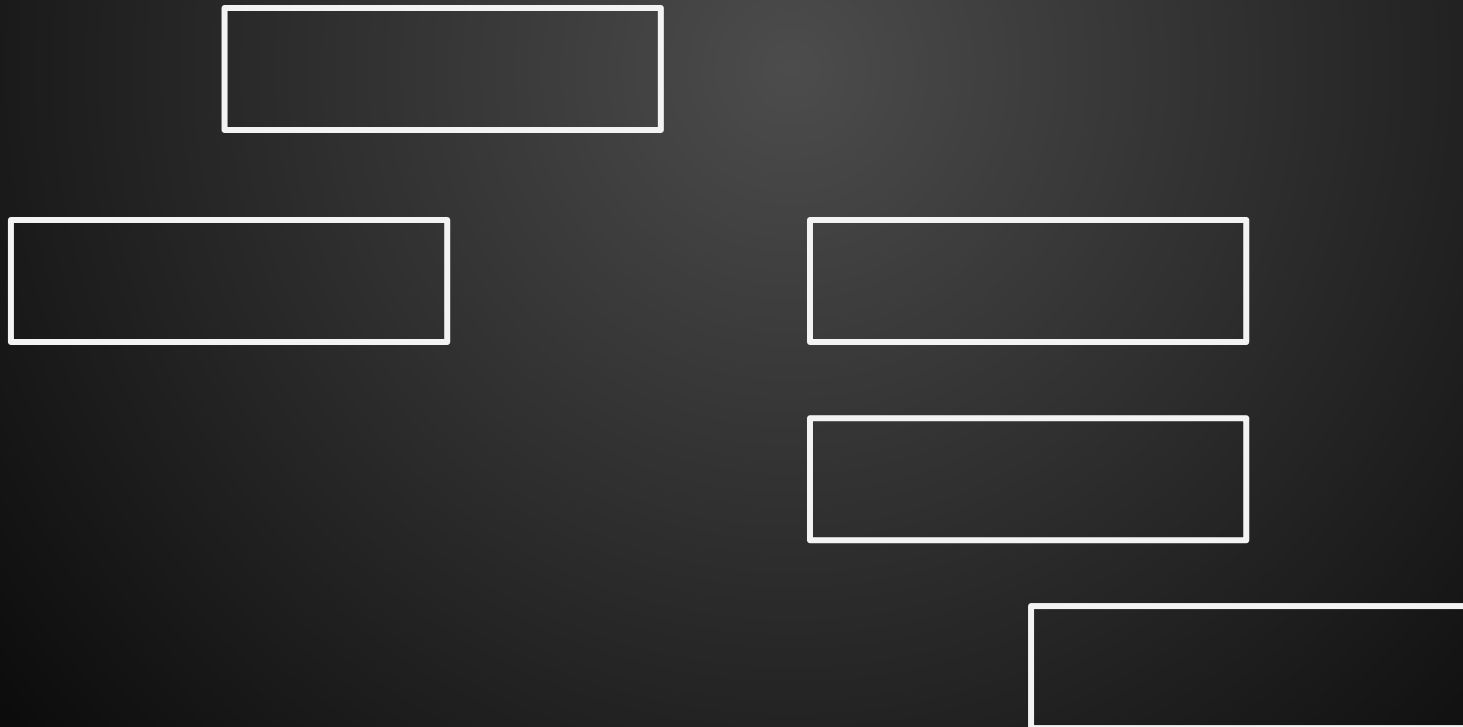
Automatisches Herunterladen und referenzieren im Erstellungsprozess

Alternative:

- Kopieren von JARs aus dem Web
- manuelles Auflösen von Versionsabhängigkeiten
- Einchecken der Bibliotheken in VCS (NEIN!!)

Abhängigkeitsmanagement

- Auflösen von Versionskonflikten
- Auflösen indirekter Abhängigkeiten



Verteilungsmanagement

- Code anderen zur Verfügung stellen und zwar nicht als GitHub-Projekt sondern als versionierte, gepackte Bibliothek
- automatisches Hochladen auf Repository-Server (Maven Central, Artifactory, Nexus, etc.)

Alternative:

- Kopieren von Quelldateien, manuelles Bauen, File Shares, etc.

Testausführung

- Nachvollziehbare Ausführung der Tests
- Unterscheidung von Testphasen
- Erstellung von Testreports zur Auswertung

Alternative:

- Manuelle Ausführung der Tests oder Verwendung irgendwelcher selbst geschriebenen Skripte :-(

Projektmanagement

Keine Gantt-Charts, Projektablaufpläne oder Meilensteine sondern:

- Management des lokalen kleinen Entwicklungsprojekt mit seinen Abhängigkeiten, Ordnern und Dateien
- Vorgabe einer Verzeichnisstruktur für Quelltext, Ressourcen und übersetzten Code

Alternative:

- manuell, irgendwie gestricktes Projektlayout

Deployment

Bereitstellung eines Weges um lokal einen Container hoch zu fahren (Jetty/Tomcat, etc.) um die Anwendung auszuführen

Alternative:

- Paketiertes WAR-Archive oder andere Artefakte manuell in einen lokal installierten Tomcat kopieren :-)
- Container manuell hoch und runter fahren auch für Tests etc. :-)

Build Tools - Übersicht

- Maven (Java/Groov)
- Gradle (Java/Groovy)
- Ant (Java/Groovy)
- Grunt (JavaScript)
- Make (C/C++)
- Rake (Ruby)
- SBT (Scala)
- Leiningen (Clojure)

Was will ich von Ihnen?

- Einsatz eines Build-Tools ist verpflichtend
- Software lässt jederzeit auf jeder Maschine bauen lassen (Voraussetzung: Java / BuildTool sind installiert)
- Abhängigkeiten werden automatisch aufgelöst
- Quelltext wird übersetzt
- Ressourcen werden durch das Build-Tool kopiert (keine manuellen Eingriffe nötig)