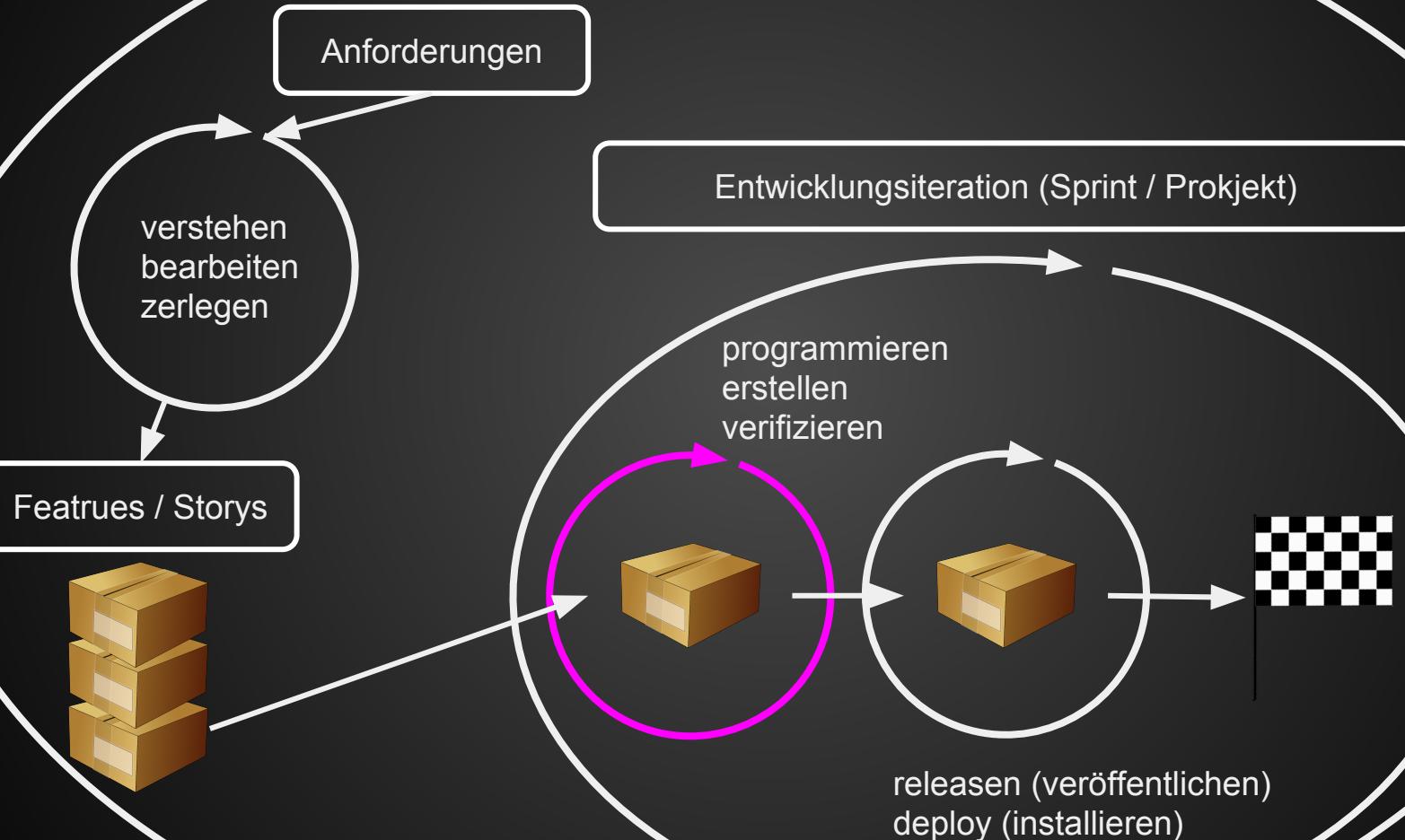


Projektlayout

Projektverzeichnisstruktur

Produktentwicklungszyklus



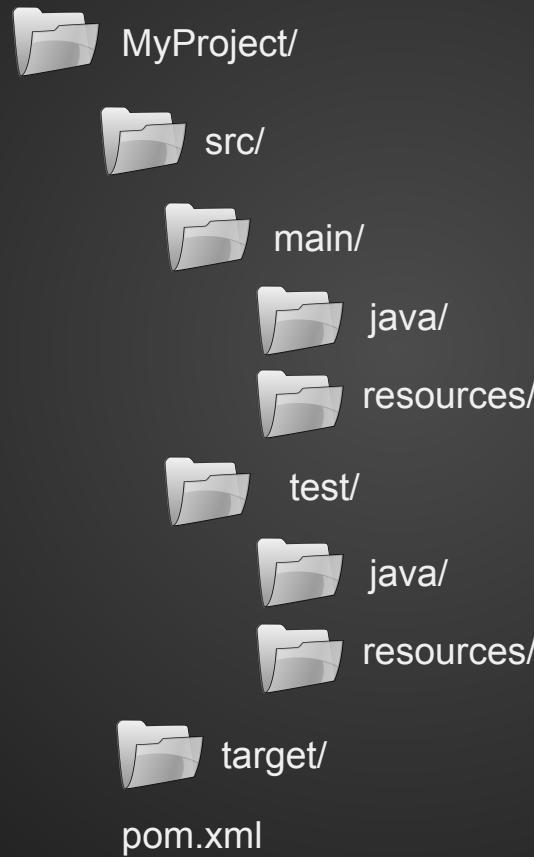
Warum ist das wichtig?

- Trennung Quellen, übersetzter Code und Ressourcen
- leichteres Zurechtfinden in verschiedenen Projekten
- Trennung zwischen Test- und Produktionscode

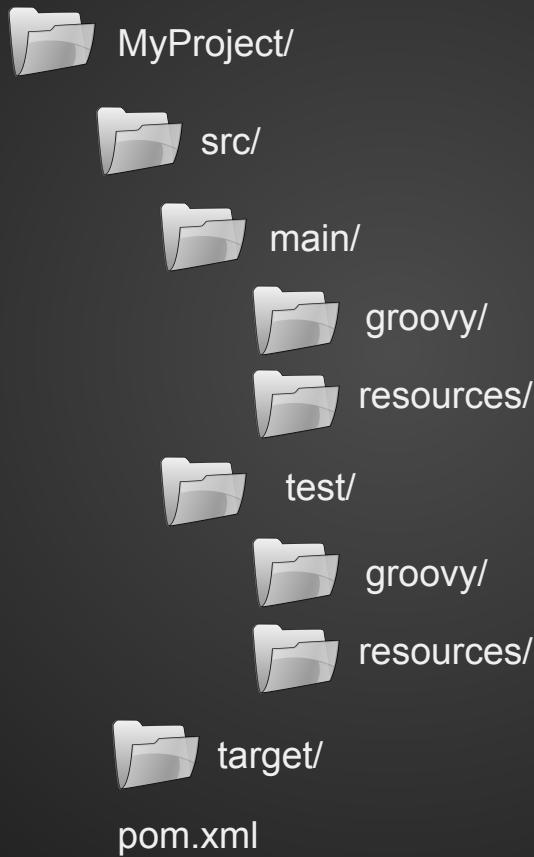
Alternative:

- Jedes Projekt = anderes Layout
- Mischung Kompilat, Quellen und Ressourcen

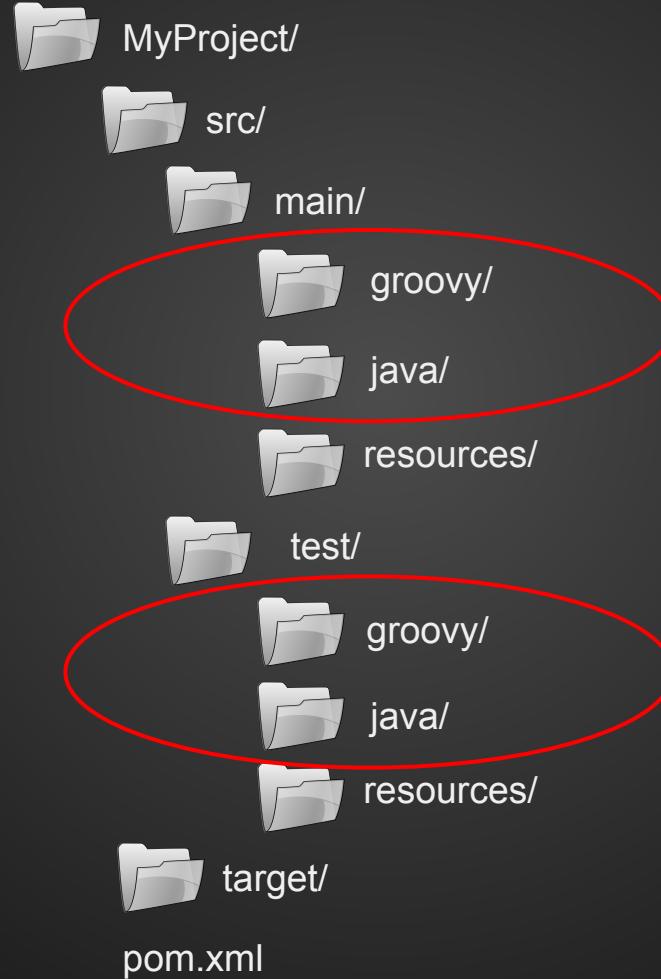
Maven-Java-Projektlayout



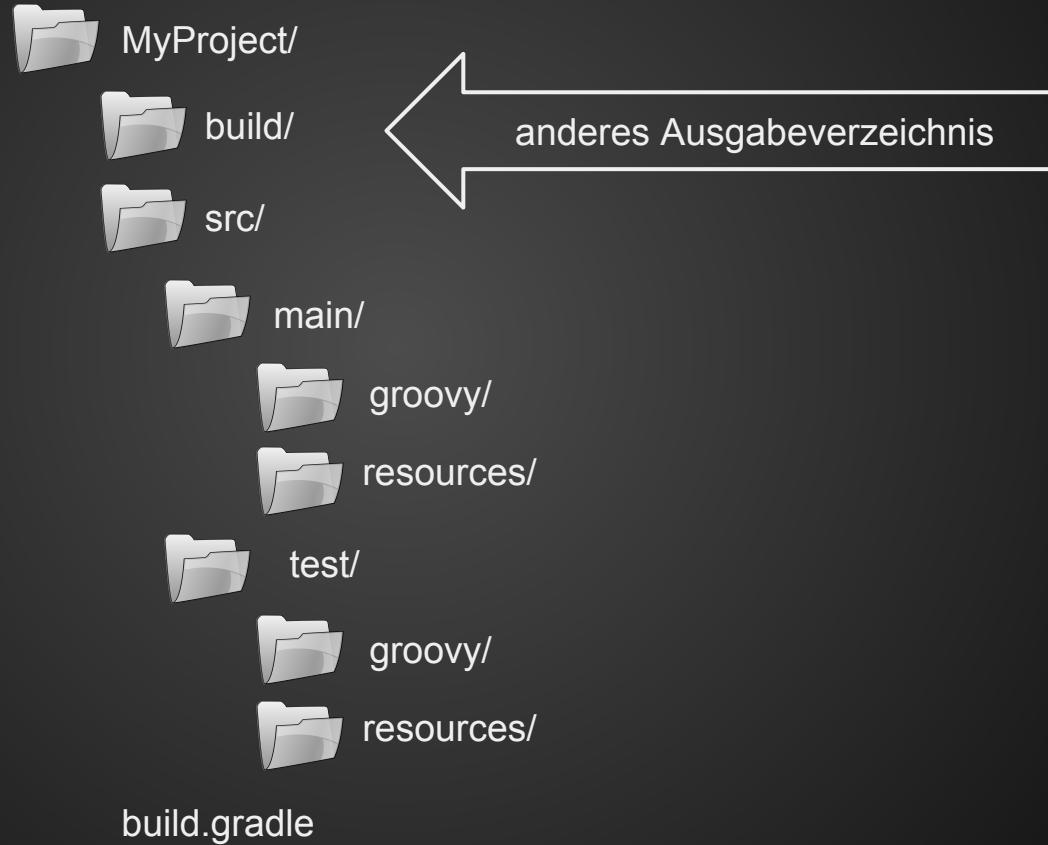
Maven-Groovy-Projektlayout



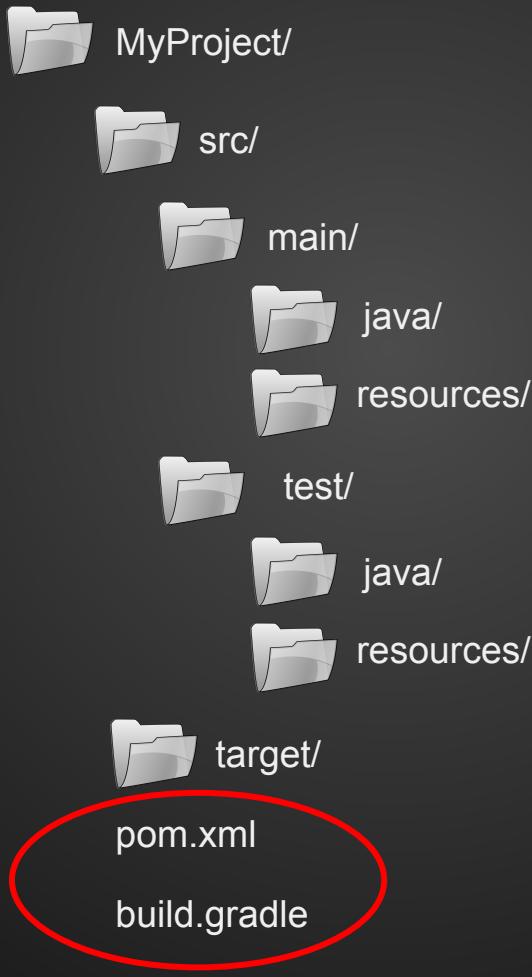
Java- und Groovy-Maven-Projektlayout



Gradle-Standard-Projekt-Layout



Maven-Gradle-Java-Projektlayout (!)



Maven-Phase zu Order

- process-resources
 - **compile**
 - process-test-resources
 - **test-compile**
 - **test**
 - **package**
 - install
 - deploy
- 
- | | |
|-------------------------------------|------------------------------|
| src/main/resources | [Icon: Folder with document] |
| src/main/java / target/classes | [Icon: Folder with document] |
| src/test/resources | [Icon: Folder with document] |
| src/test/java | [Icon: Folder with document] |
| src/test/java / target/test-classes | [Icon: Folder with document] |
| target/classes / target/resources | [Icon: Folder with document] |
| \$HOME/.m2 | [Icon: Folder with document] |
| [Hochladen auf Repository Server] | [Icon: Folder with document] |

http://maven.apache.org/guides/introduction/introduction-to-the-lifecycle.html#Setting_Up_Your_Project_to_Use_the_Build_Lifecycle

Archetypes und Projekt-Templates

Maven und Gradle haben Plugins um Projektvorlagen mit entsprechender Verzeichnisstruktur zu generieren

Beispielsweise für:

- einfache App
- Web-App
- J2EE
- etc.

Maven Archetype Plugin

```
$ mvn archetype:generate
```

oder

```
$ mvn archetype:generate -Dfilter=groovy
```

GRADLE:

```
$ gradle setupBuild --type java-library
```

- mit Enter Vorgabe bestätigen oder ändern
- generiert Verzeichnisse und pom.xml

Gradle Build Setup Plugin

```
$ gradle setupBuild --type java-library
```

generiert:

- build.gradle
- settings.gradle
- Verzeichnisstruktur

Gradle Build Setup Plugin

Typen:

- pom - Konvertiert pom.xml zu build.gradle
- basic - Erzeugt settings + build.gradle
- java-library - Erzeugt gradle-Projekt
einschließlich der Verzeichnissstruktur

=> Ist noch experimentell!

Gradle Templates Plugin

build.gradle erweitern:

```
apply from: 'http://www.tellurianring.  
com/projects/gradle-plugins/gradle-  
templates/apply.groovy'
```

```
$ gradle tasks
```

```
$ gradle createGroovyProject
```

Fragen?