Wprowadzenie do systemu budowania aplikacji Maven (i Gradle)

Mateusz Haligowski
15 kwietnia 2014

Cześć, jestem Mateusz Haligowski

- na co dzień programuję w **Javie**, prywatnie nie tylko (choć głównie JVM)
- pracuję w **JIT Solutions** przy projektach w **Young Digital Planet**
- kończyłem ekonometrię ze statystyką oraz informatykę
- udzielam się głównie na **GitHubie** (mhaligowski)

Opowieść z pola boju #1

Mój pierwszy raz

Krótka historia narzędzi do budowania

- C/C++: kompilacja -> linkowanie -> aplikacja
- Java: javac Klasa.java -> Klasa.class
- pliki shellowe
- make, GNU Automake/Autoconf
- Apache Ant

O Apache Ant słów kilka

```
…
<target name="init">
  <!-- Create the time stamp -->
  <tstamp/>
  <!-- Create the build directory structure used by compile -->
  <mkdir dir="${build}"/>
</target>

<target name="compile" depends="init"
      description="compile the source " >
  <!-- Compile the java code from ${src} into ${build} -->
  <javac srcdir="${src}" destdir="${build}"/>
</target>
…
```

Opowieść z pola boju #2

Apache Maven

- project management tool - więcej niż Ant
- rozwiązywanie zależności
- wprowadzony cykl życia projektu

Zarządzanie projektem

- Wszystkie informacje o projekcie: autorzy, zarządzanie kodem
- Budowanie projektu
- Publikowanie projektu
- Generowanie dokumentacji

Instalacja Maven-a

- http://maven.apache.org
- rozpakować i dodać do PATH-a
- przydaje się ustawienie zmiennych środowiskowych
- uruchamianie przez **mvn [polecenie]**

Plik pom.xml

```
<project>

  <modelVersion>4.0.0</modelVersion>

  <groupId>com.mycompany.app</groupId>

  <artifactId>my-module</artifactId>

  <version>1</version>

</project>
```

Struktura katalogów wg Mavena

src/main/java	źródła projektu
src/main/resources	zasoby projektu
src/main/webapp	pliki aplikacji webowej (web.xml itp.)
src/test/java	testy
src/test/resources	zasoby do testów

Archetypy

maven-archetype-quickstart	prosty projekt, tylko struktura katalogów
maven-archetype-webapp	aplikacja typu WAR
maven-archetype-j2ee-simple	aplikacja JEE

Generowanie archetypu

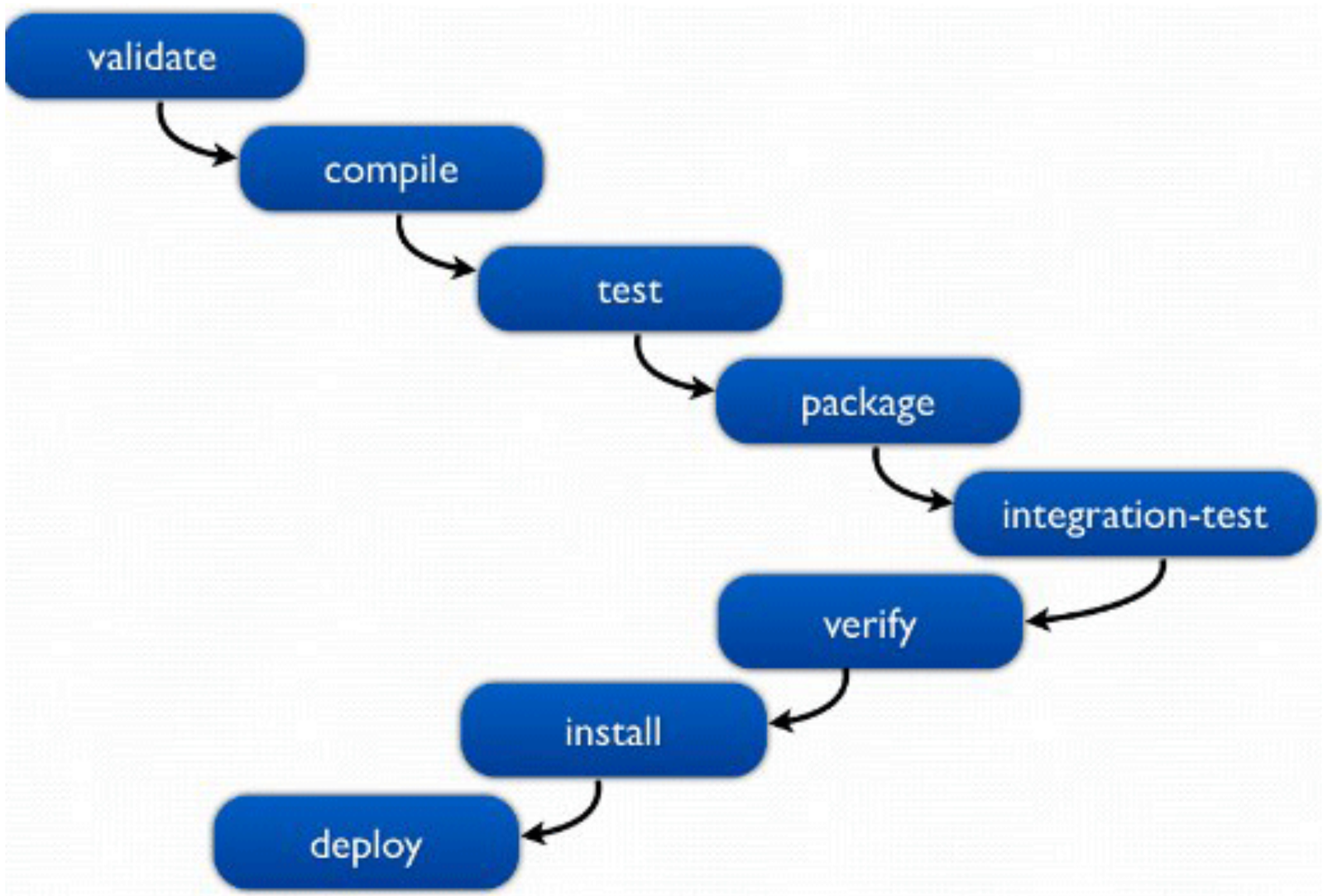
```
mvn archetype:generate \

-DarchetypeGroupId=org.apache.maven.archetypes \

-DgroupId=com.mycompany.app \

-DartifactId=my-app
```

Build lifecycle



Pluginy a cykl życia Mavena

clean	maven-clean-plugin
process-sources	maven-dependency-plugin
compile	maven-compiler-plugin
test	surefire
package	maven-jar-plugin/maven-war-plugin

Popularne pluginy

- maven-ant-plugin
- maven-surefire-plugin
- maven-checkstyle-plugin
- gwt-maven-plugin
- jetty-maven-plugin

Uruchamianie Mavena

1. Przez fazę (fazy) cyklu życia: **mvn install**

2. Przez wtyczkę: **mvn jetty:run**

Zależności

- wsparcie dla zarządzania zależnościami
- możliwość zdefiniowania odpowiedniego zakresu (compile, provided, test, system…)
- UWAGA: nie chroni przed powtarzającymi się bibliotekami (transitive dependency)

Małe demo

- projekt spring-petclinic

O czym jeszcze warto wiedzieć?

- subprojekty
- profile
- repozytoria

Dlaczego Gradle?

- technologia opracowana przez firmę Gradleware
- obsługa wielu różnych technologii
- bardziej wydajne od Mavena
- szybszy rozwój

Cechy gradle-a

- zaimplementowany w Javie, z wierzchnią warstwą w Groovym
- skutkiem tego plik build.gradle jest tak naprawdę skryptem w Groovy - nie definiujesz tego jak twój projekt ma być zbudowany, tylko to oprogramowujesz!

Instalacja

A. podobnie jak Maven - www.gradle.org

B. Gradle Wrapper

Demo

- z naszego rzeczywistego projektu

Co jeszcze w Gradle?

- Gradle for Android

Pytania?