



Apache Maven

Trener: Marcin Górecki

Agenda

- 9:00 – 10:30 Podstawy - po co jest Maven, dlaczego go używamy
- 10:30 – 10:45 Przerwa
- 10:45 – 12:30 Przykłady i omówienie ćwiczeń
- 13:30 – 15:00 Ćwiczenia

Definicja, która na początku nic nie wyjaśnia

Maven, a Yiddish word meaning *accumulator of knowledge*, was originally started as **an attempt to simplify the build processes** in the Jakarta Turbine project. There were several projects each with their own Ant build files that were all slightly different and JARs were checked into CVS.

We wanted a standard way to build the projects, a clear definition of what the project consisted of, an easy way to publish project information and a way to share JARs across several projects.

Maven jest wszędzie

Praktycznie niemożliwe we współczesnej firmie jest tworzenie dobrej jakości oprogramowania bez automatyzacji procesu budowania, reużywalności kodu, automatyzacji testów etc.



Google Play

Nie piszecie sami aplikacji na swoje telefony

Jakich „appek” potrzebuje firma w swojej aplikacji

Moduł dostępu do bazy danych (ORM)

Moduł obliczeniowy matematyczne dostarczony przez R&D

Obsługa MVC i servletów

Moduł logowania

Klient obsługi płatności

Webservices

Wsparcie testowania

Code candies

Zależności

```
<dependency>
    <groupId>commons-collections</groupId>
    <artifactId>commons-collections</artifactId>
</dependency>

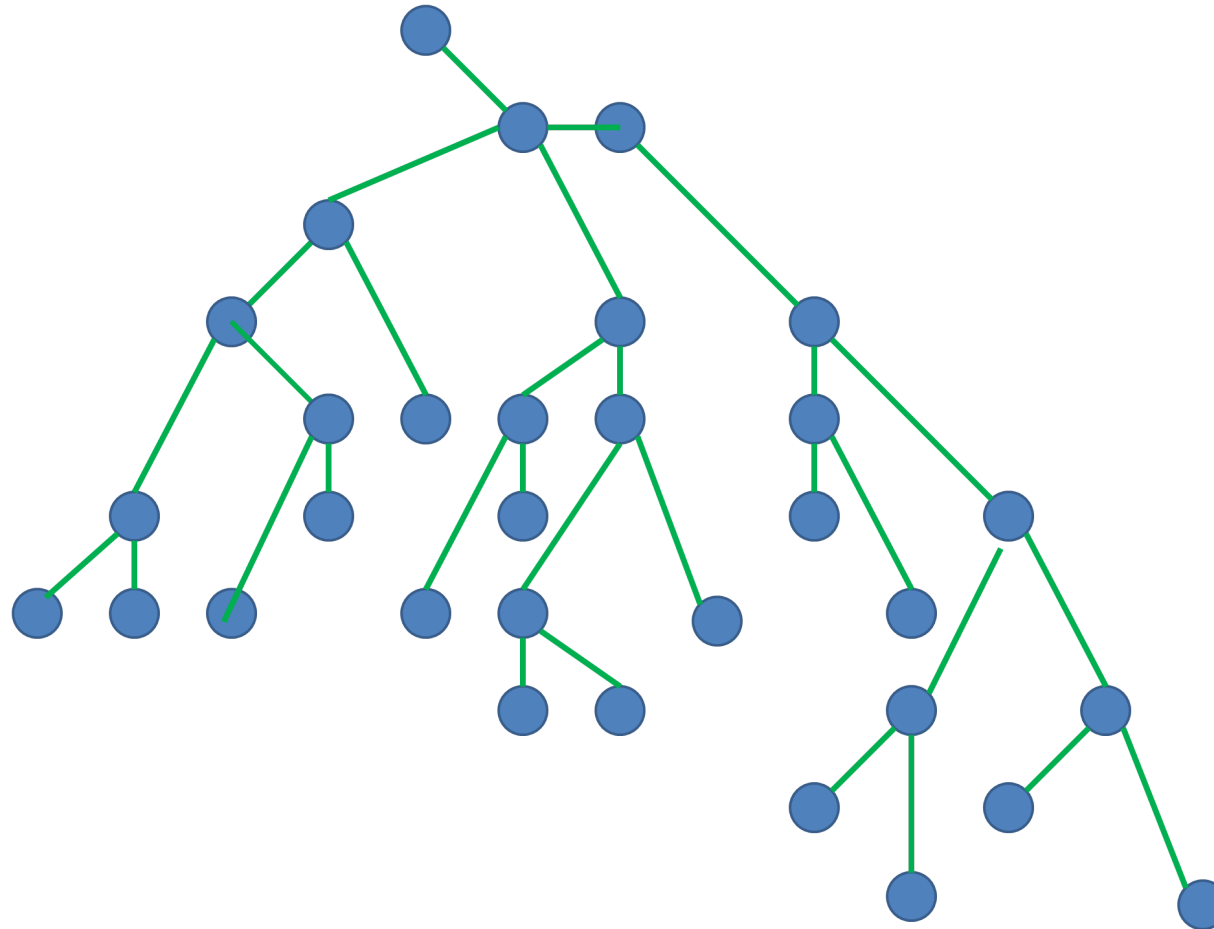
<dependency>
    <groupId>joda-time</groupId>
    <artifactId>joda-time</artifactId>
</dependency>

<dependency>
    <groupId>log4j</groupId>
    <artifactId>log4j</artifactId>
</dependency>

<dependency>
    <groupId>net.sourceforge.javacsv</groupId>
    <artifactId>javacsv</artifactId>
</dependency>

<dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-annotations</artifactId>
</dependency>
```

Jak wyglądają prawdziwe aplikacje

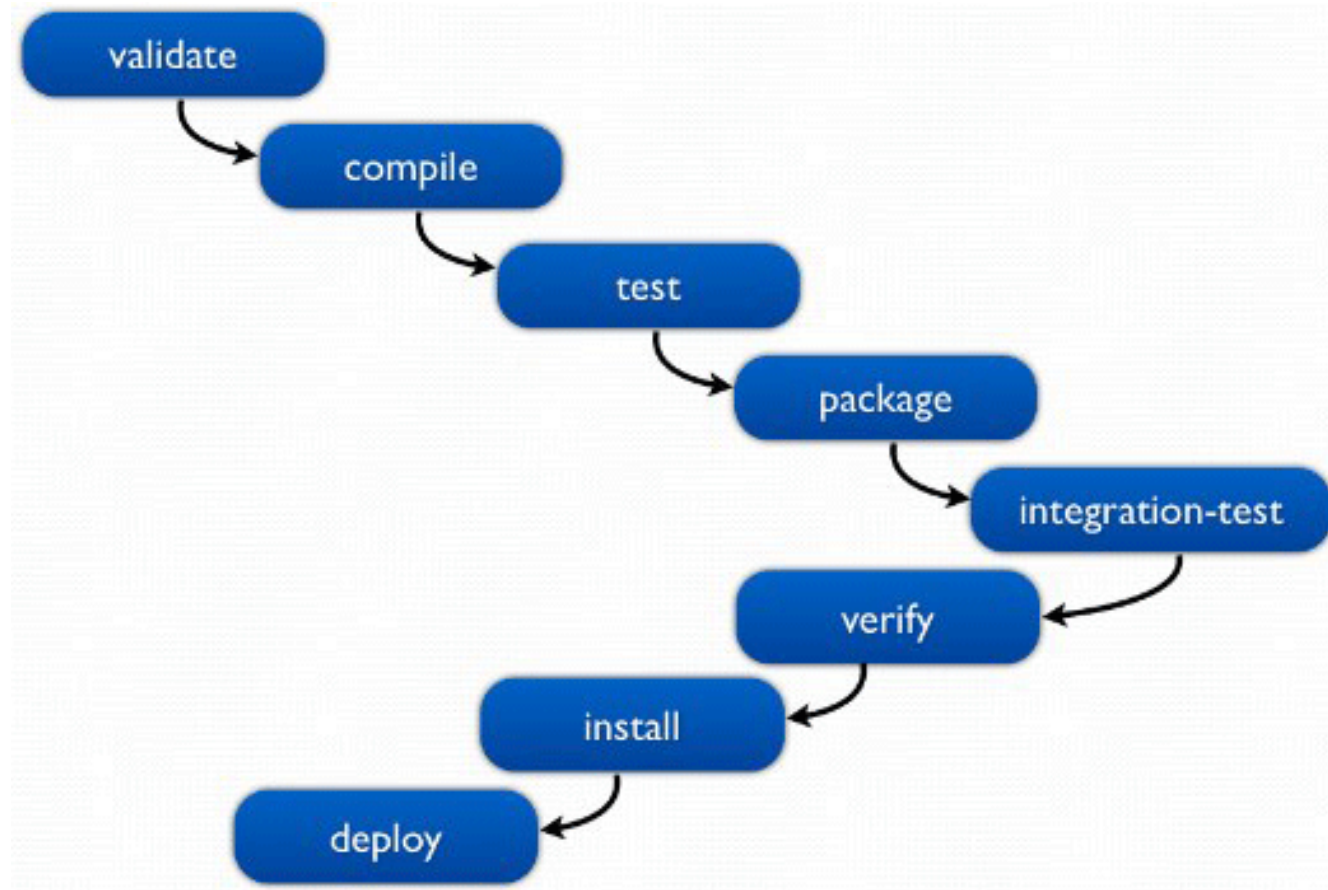


Jak wyglądają prawdziwe aplikacje

- Korzystają z modułów innych programistów
- Zależności między modułami aplikacji
- DRY – Don't Repeat Yourself
- Generowanie kodu (np. REST)
- Różne wersje aplikacji dla środowisk DEV/PROD - profile
- Quality assurance – Automatyczne uruchamianie testów
- Continuous integration



„Taśma produkcyjna” albo Project Lifecycle



Project Lifecycle

- *mvn validate* : czy dobrze zdefiniowaliśmy jak chcemy budować samochód
- *mvn compile* : ściągamy podzespoły od dostawców, składamy gotowe części w całość
- *mvn test* : ściągamy od dostawców pomoce do testowania (np. mierniki poziomu spalin, mierniki grubości lakieru, wyważarki kół), budujemy narzędzia do sprawdzania jakości pojedynczych podzespołów (czy kręcą się koła etc.)
- *mvn package* : przygotowujemy gotowy produkt (naklejamy folie ochronne, wkładamy w opakowanie)

Project Lifecycle

- *mvn verify* : testy integracyjne – czy podzespoły ze sobą prawidłowo współpracują – uruchamiamy silnik i próbujemy się przejechać
- *mvn install* : wysyłamy gotowy produkt do magazynu naszej firmy, tak aby był dostępny do użytku wewnętrznego (UWAGA: tylko na własnym komputerze). Inne działy firmy mogą użyć samochodu jako zależności
- *mvn deploy* : wysyłamy gotowy produkt do centralnego magazynu – wszystkie firmy mogą nim się przejechać, ale wiadomo że to nie gotowy produkt
- *mvn release* : produkt jest gotowy – nadajemy mu finalną wersję i wprowadzamy na rynek

To dopiero początek...