

# Maven





# Maven

- ▶ Why?
- ▶ Definition
- ▶ Core concepts
- ▶ Artifact
- ▶ Scope
- ▶ Pom.xml





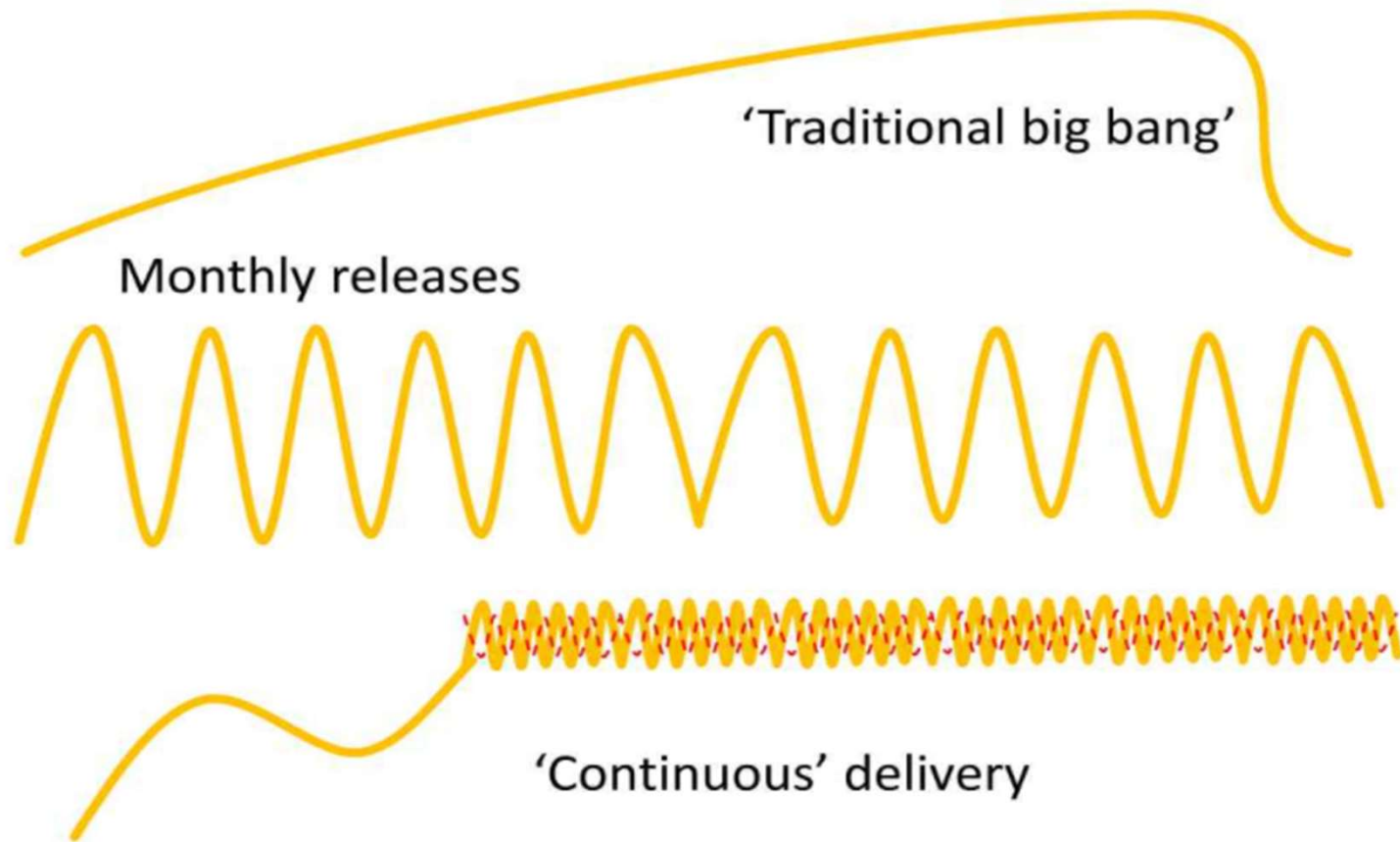
# What is Maven?

**Maven is a build automation tool for building and managing software.**



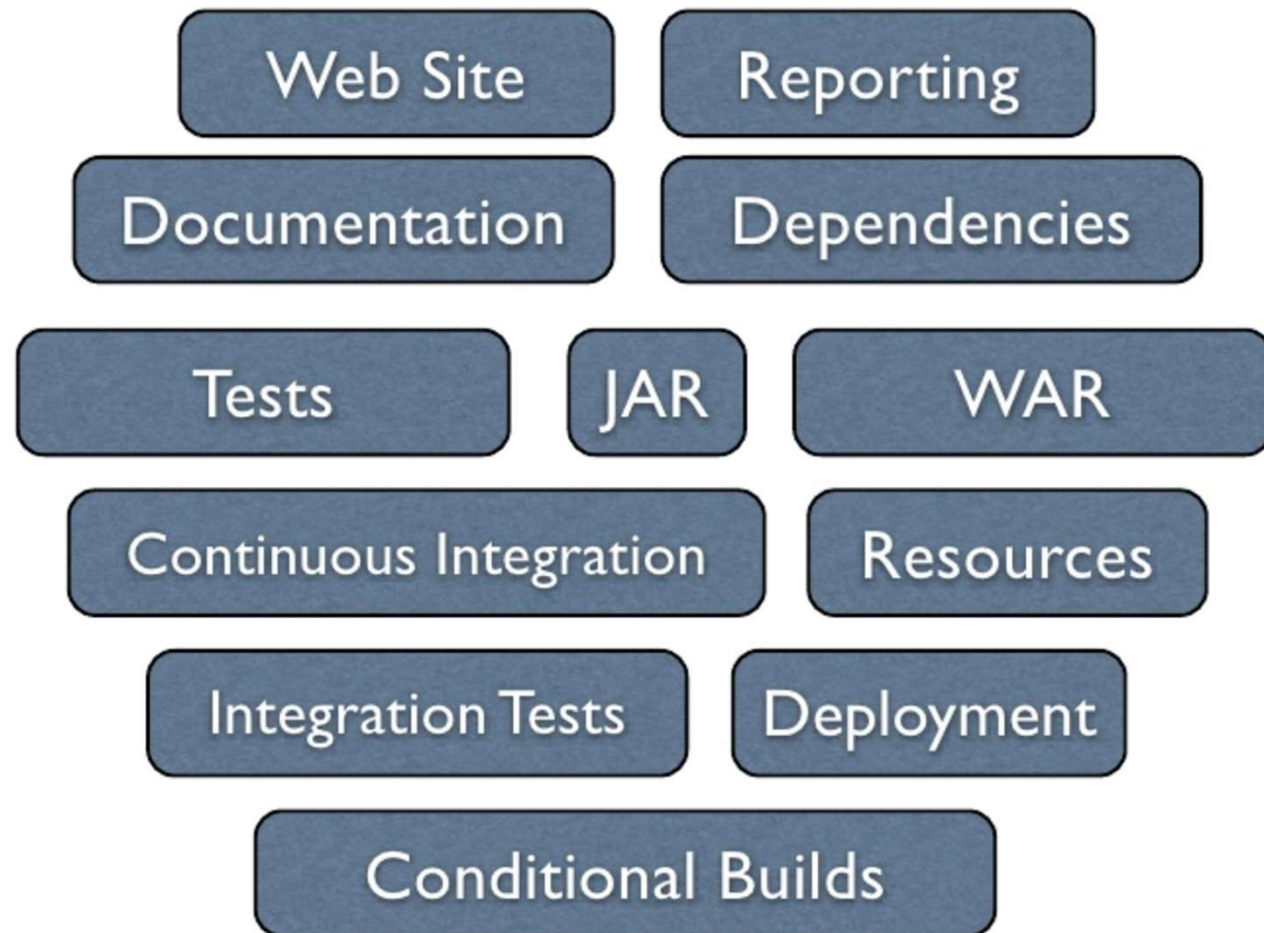
# Why?

## Evolution of Continuous Delivery





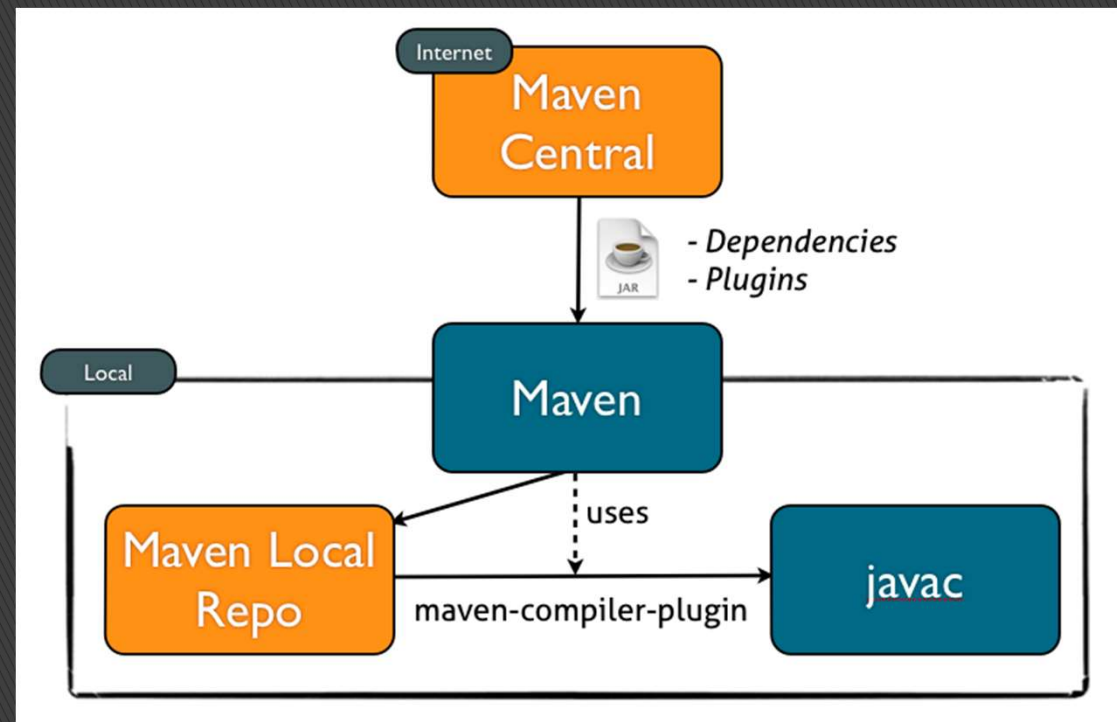
# Why?



# Maven – definition

It manages:

- ▶ Documentation
  - ▶ Testing
  - ▶ Dependencies
  - ▶ Reporting
  - ▶ Build lifecycle
- and much more...





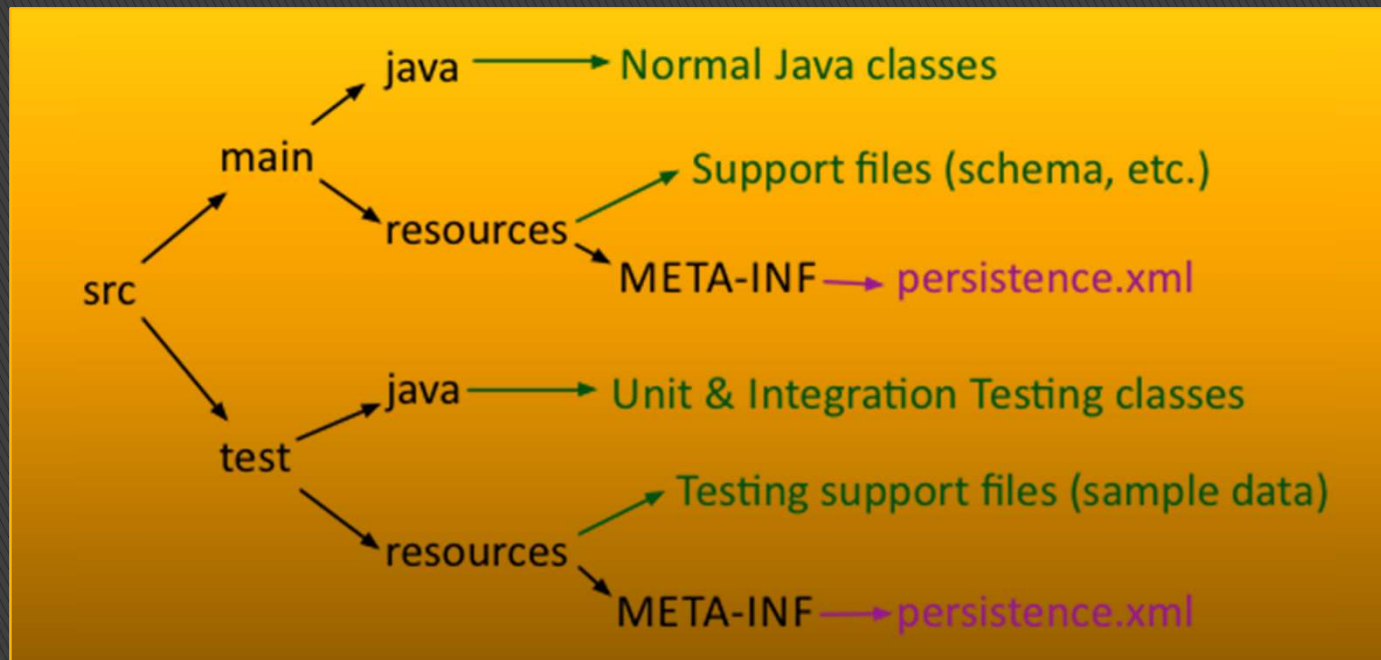
# Maven's Objectives

- ▶ Making the build process easy
- ▶ Providing a uniform build system
- ▶ Providing quality project information
- ▶ Providing guidelines for best practices
- ▶ Allowing transparent migration to new features



# Maven = Agreement on configuration structure

- ▶ Folder structure
  - Source, Tests, Resources
- ▶ Knowledge of industry standards needed





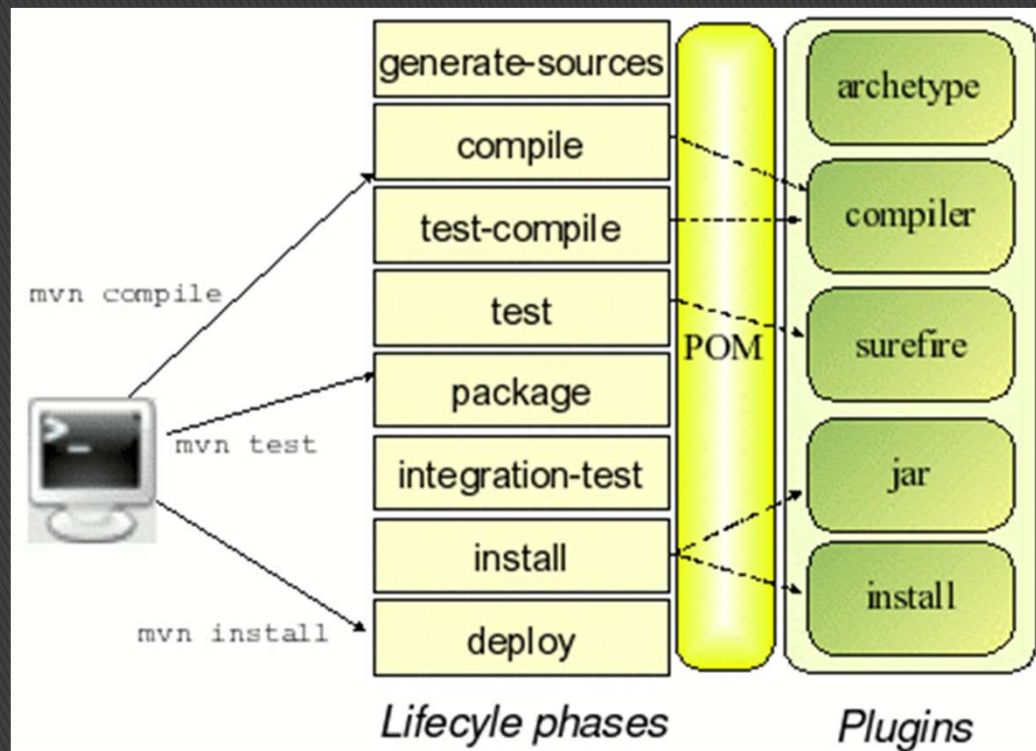
# Core concepts

- **Artifact:** unit of deployment (jar, war, ear), final result of development process
- **Repository:** library(local/remote/central), from where artifacts can be downloaded
- **Build lifecycle:** well defined number of steps (phases) with a fixed order
- **Phase:** steps of the build lifecycle: each step has some tasks (so called **goals**) which are executed during the phase
- **Goal:** a well defined task e.g. clean – removes the files produced by previous builds





# Core concepts



**Plugin:** a tool (software) build for a specific goal – each plugin has some goals it can execute

e.g. surefire: will execute the tests in the project

- compiler: compiles the Java source code
- install: places the artifact in your local repository
- deploy: deploys it to a remote server



# Maven artifact

- ▶ groupId: unique identifier of your project, a domain (org.jboss), name of the project (tools), submodules (examples)
- ▶ artifactId: name of the product (jboss-javaee6-webapp)
- ▶ version: version number
- ▶ packaging: type of packaging (war, jar, ear)

```
<groupId>org.jboss.tools.examples</groupId>  
  <artifactId>jboss-javaee6-webapp</artifactId>  
    <version>0.0.1-SNAPSHOT</version>  
  <packaging>war</packaging>
```





# Scope

Defines when and from where our dependencies can be reached:

- ▶ Compile – packaged with the project (for example in a lib folder)
- ▶ Provided – provided by the container or jdk, will not be packaged with the product
- ▶ Runtime – no reference in the code but will be used runtime (JDBC driver)
- ▶ Test – only accessible on the test classpath





# pom.xml

- ▶ Project Object Model description file
- ▶ It includes:
  - ▶ Artifact description
  - ▶ Artifact modules
  - ▶ Dependencies
  - ▶ Development profiles
  - ▶ ...

