Maven

Wrocław, 04.2021, Marcel Rzepka

What is maven?

Apache Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.

You can use it for:

- Building project
- Deployment (jar, war)
- Dependency management
- Running your application
- Testing your application on multiple levels
- Generating documentation

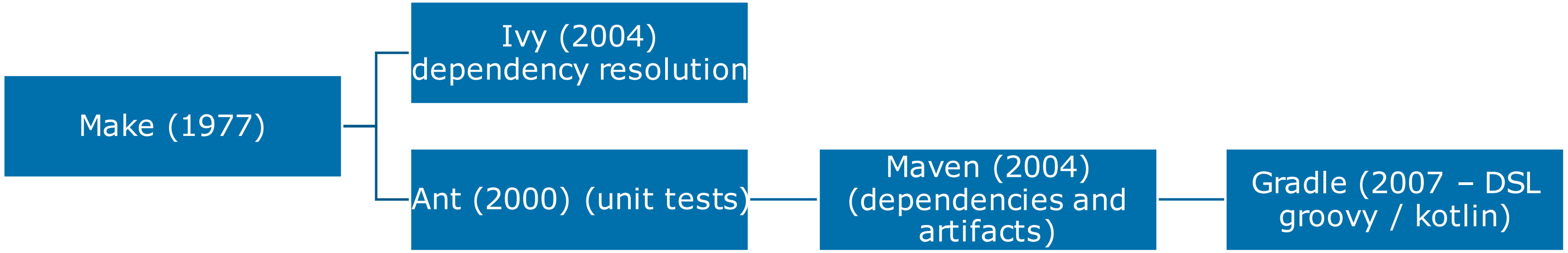
Its just control panel?

Context

Why not just use gcc or javac?

https://github.com/git/git/blob/master/Makefile

Context


Make (1977)
Ivy (2004) dependency resolution
Ant (2000) (unit tests)
Maven (2004) (dependencies and artifacts)
Gradle (2007 – DSL groovy / kotlin)

https://kotlinfrompython.com/2019/02/07/gradle-maven-ant-other-build-tools-and-history/

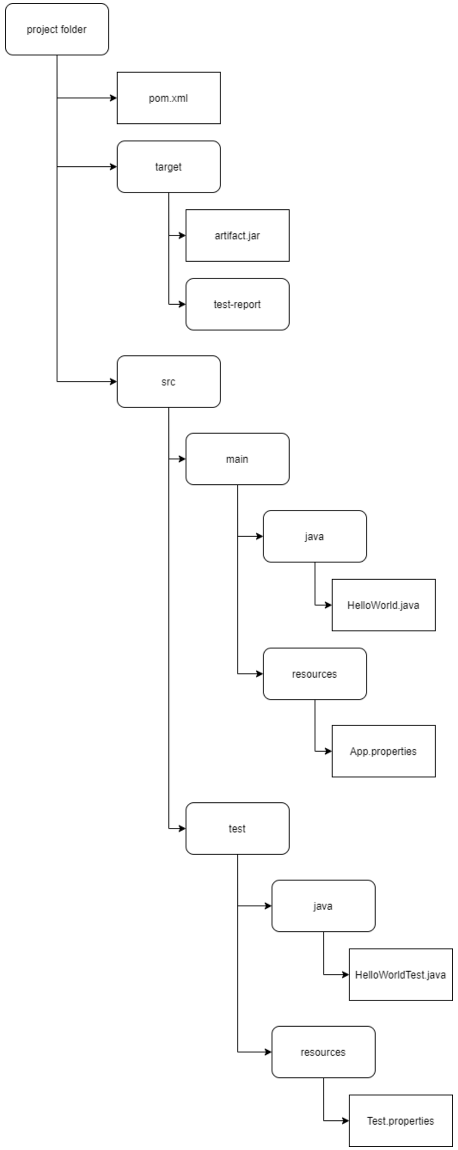
Context

Too complicated.

- Build variables
- System dependencies such as CURL management
- Installation / packaging

Project structure

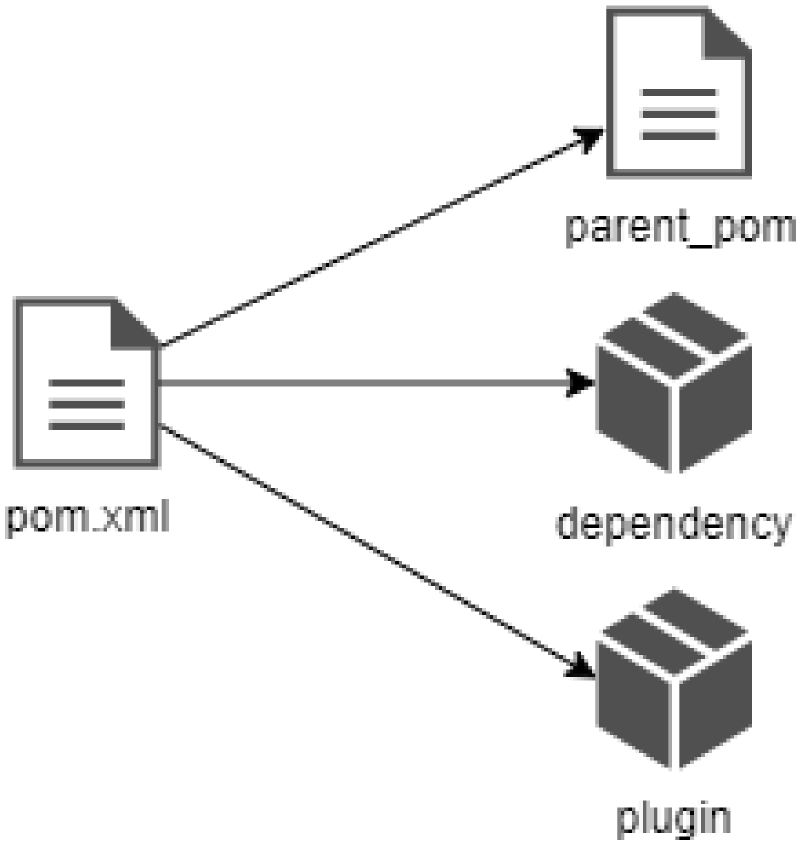
Directories



https://maven.apache.org/guides/introduction/introduction-to-the-standard-directory-layout.html

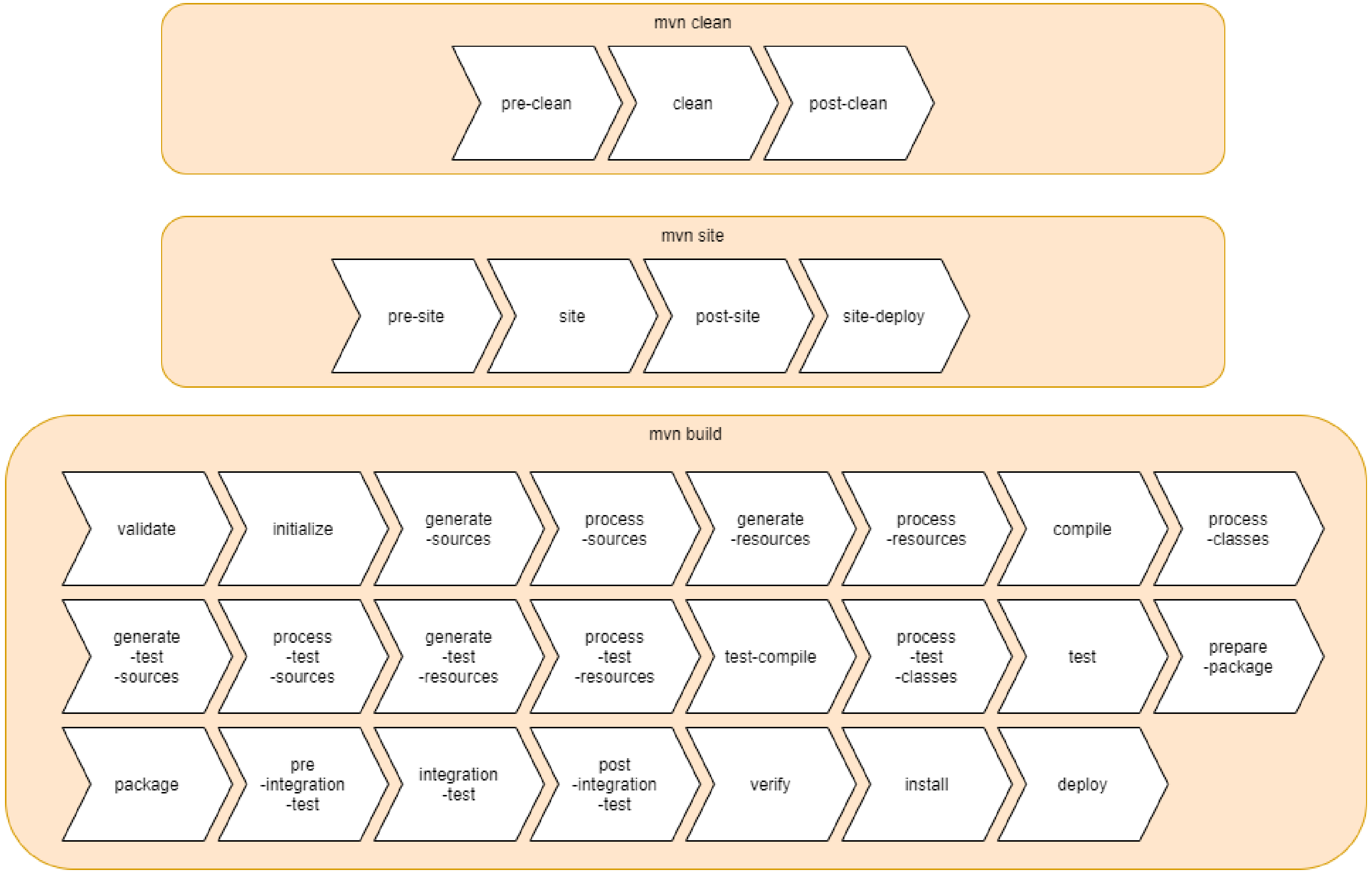
Project structure

Logical



Lifecycle

https://maven.apache.org/guides/introduction/introduction-to-the-lifecycle.html



Dependency vs Plugin

- Plugin
  - extends maven possibilities –> e.g. packing artifact's to zip
- Dependency
  - extends java project which is builded –> e.g. apache strings commons – additional classes for handling strings

Usefull commands

- mvn command1 command2 … commandN

- mvn clean – cleans the target folder
- mvn compile test-compile – compiles the code
- mvn test – runs test
- mvn verify – run integration test
- mvn install – install to local repository
- mvn deploy – deploy code to remote repository

- mvn dependnecy:tree –Dverbose – genarate dependnecy tree
- mvn help:evaluate –Dexpression=project.version – evaluate expresion; show version

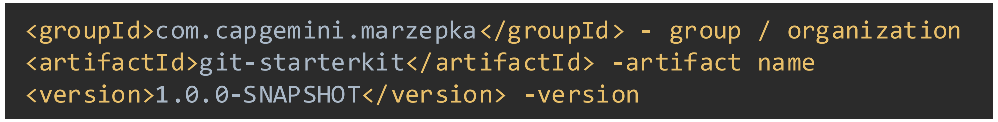
Maven .m2

Local repository

Contains your dependencies

Maven pom

https://maven.apache.org/pom.html



```
<groupId>com.capgemini.marzepka</groupId> - group / organization
<artifactId>git-starterkit</artifactId> -artifact name
<version>1.0.0-SNAPSHOT</version> -version
```

Conflict?

http://maven.apache.org/plugins/maven-dependency-plugin/examples/resolving-conflicts-using-the-dependency-tree.html

Real pom
https://github.com/apache/commons-io/blob/master/pom.xml

Maven settings

https://maven.apache.org/settings.html

- Location of .m2 folder
- Servers – credentials repositories (password encryption https://maven.apache.org/guides/mini/guide-encryption.html)
- Proxies, Mirrors
- Profiles

Repository

There are many public maven reposiotries but most known is: https://mvnrepository.com/repos/central

We can create our private repository:

- Nexus
- Artifactory

Versioning

Semantic Versioning

- https://semver.org/
- 3 numbers major.minor.patch
- MAJOR version when you make incompatible API changes,
- MINOR version when you add functionality in a backwards compatible manner, and
- PATCH version when you make backwards compatible bug fixes.

https://semver.org/ https://0ver.org/

Versioning

Zero versioning

- https://0ver.org/
- No major version – it’s always 0
- Major version zero (0.y.z) is for initial development. Anything MAY change at any time. The public API SHOULD NOT be considered stable. (semver.org)
- Every version might bring breaking changes
- React Native; axios; hugo; bitcoin

https://semver.org/ https://0ver.org/

Links

https://www.jrebel.com/blog/maven-cheat-sheet