CS 342 Software Design

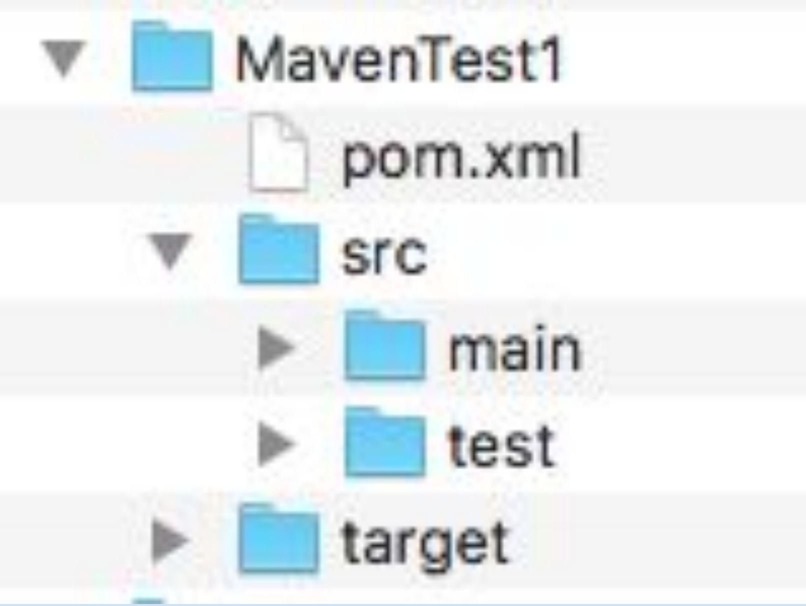
Configuration as code: Maven

What is Maven?

- Makes the build process easy with a uniform build system.
- Developers can use the environment of their choice.
- Provides standard documentation for project development.
    - Unit test reports, dependency lists
- Testing source code is parallel to project code.

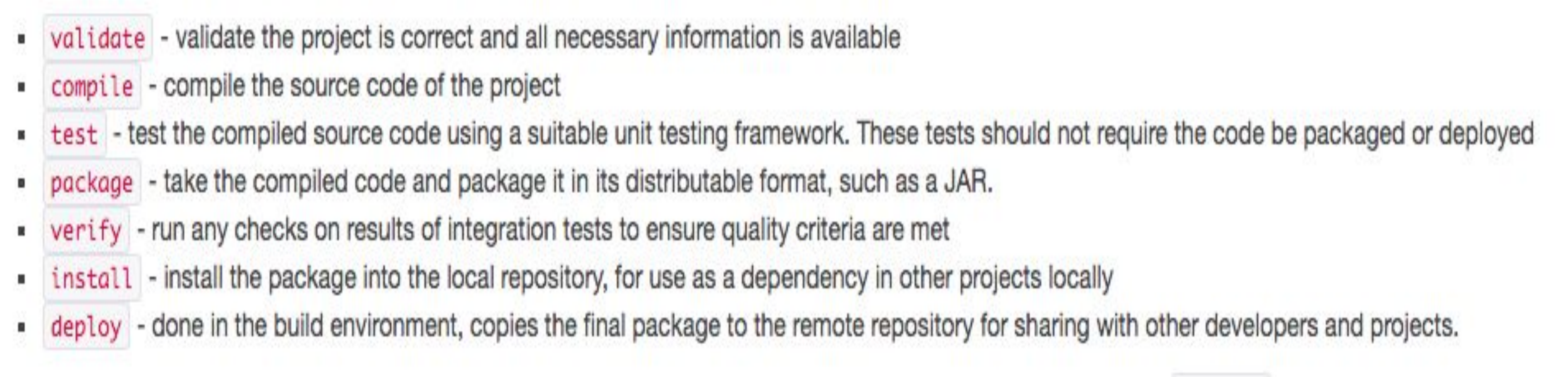
Maven POM file: Project Object Model

Xml file that contains info on project and configuration details used by Maven to build the project.



```
▼ MavenTest1
      pom.xml
  ▼ src
    ▶ main
    ▶ test
  ▶ target
```

Maven build lifecycle:



- `validate` - validate the project is correct and all necessary information is available
- `compile` - compile the source code of the project
- `test` - test the compiled source code using a suitable unit testing framework. These tests should not require the code be packaged or deployed
- `package` - take the compiled code and package it in its distributable format, such as a JAR.
- `verify` - run any checks on results of integration tests to ensure quality criteria are met
- `install` - install the package into the local repository, for use as a dependency in other projects locally
- `deploy` - done in the build environment, copies the final package to the remote repository for sharing with other developers and projects.

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

Transitive Dependencies

No need to track down the dependencies of your dependencies. Maven does this for you.

Maven: Command Line and Eclipse

Let’s take a look at how to create a Maven project on both the command line and eclipse.