Maven

Build automation and software project management tool

What are the steps involved in project deployment?

1. Check-in the code from all projects in progress into source code repository
2. Download the complete source code from source code repository
3. Build the application.
4. Store the build output either WAR or EAR file to a common network location.
5. Get the file from network and deploy the file to the production site.
6. Updated the documentation with date and updated version number of the application.

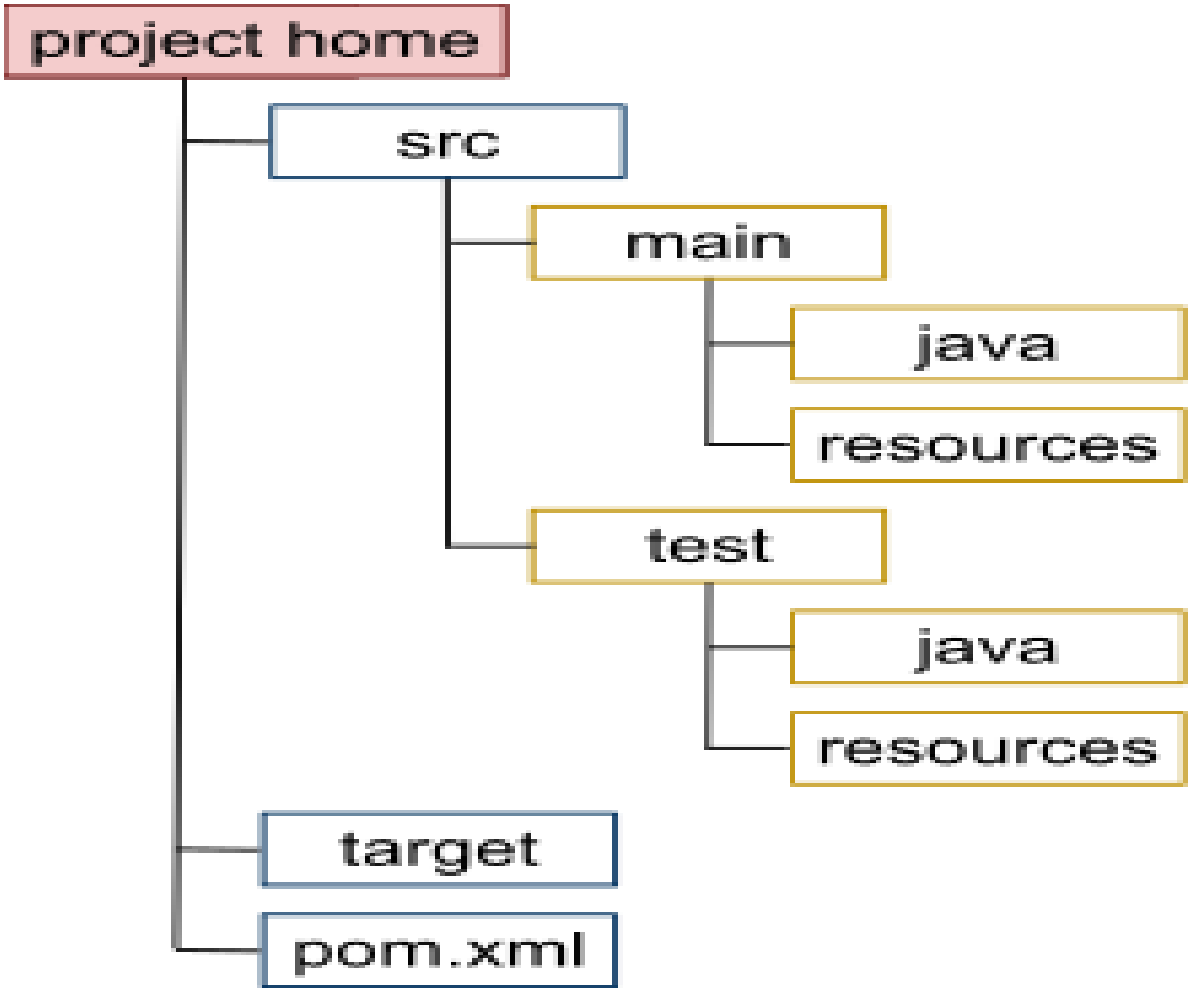
Problems without using Maven

1. Adding set of Jars in each project
2. Creating the right project structure
3. Building and Deploying the project

What is Maven??

- Maven is a project management , build automation and comprehension tool
- It provides developers a complete build lifecycle framework
- Development team can automate the project's build infrastructure
- Maven uses a standard directory layout and a default build lifecycle
- Convention over configuration
  - sensible default behavior for projects (default project structure)
  - pom.xml to define any configuration

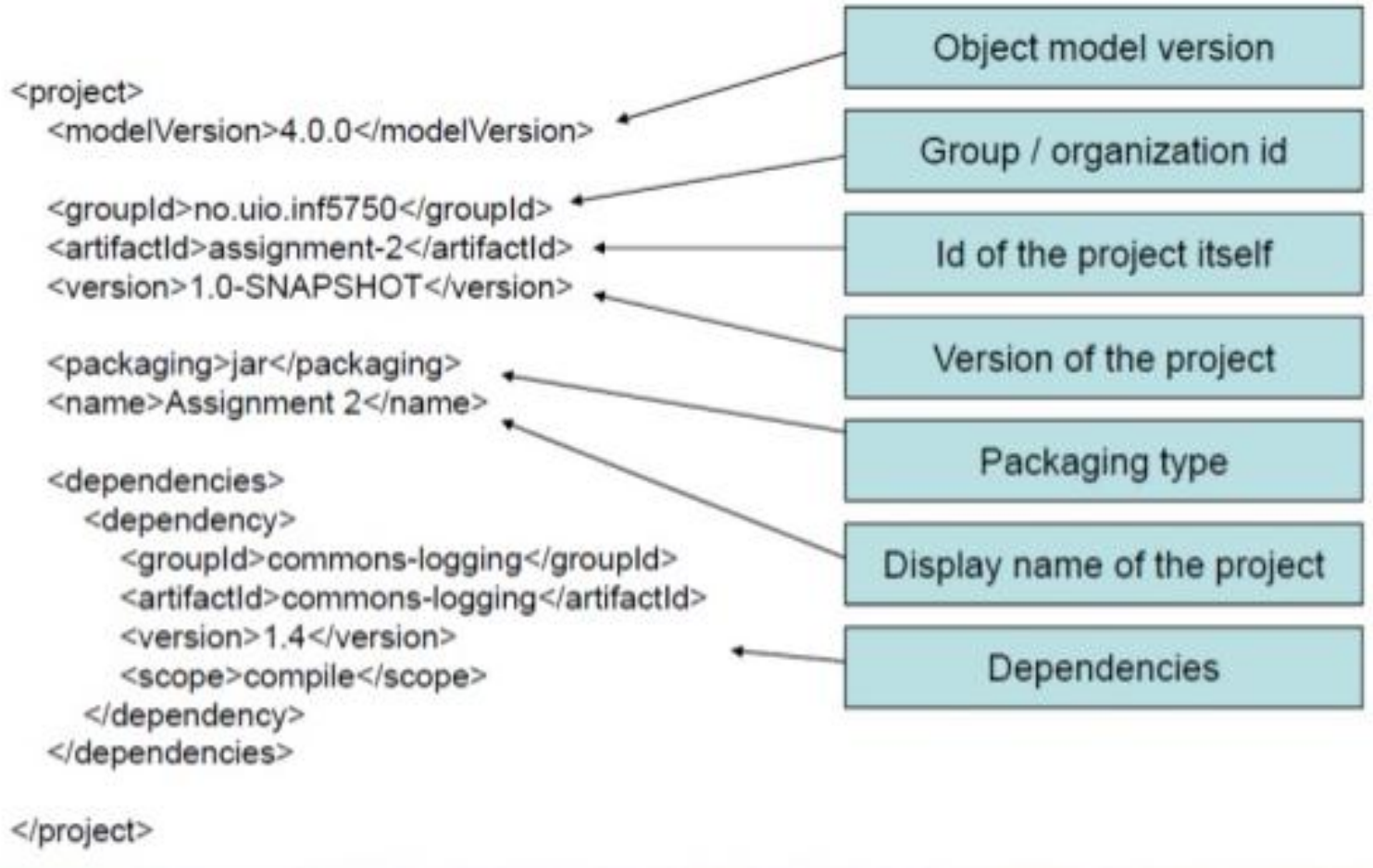
Maven Default Folder Structure



POM.xml

- POM stands for Project Object Model
- Fundamental Unit of Work in Maven and is an XML file.
- It contains information about the project and various configuration details used by Maven to build the project(s).
- Some of the configuration that can be specified in the POM are following:
  - project dependencies
  - plugins
  - goals
  - build profiles
  - project version
  - developers

What information does POM contain?



Maven 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.

Maven Lifecycle

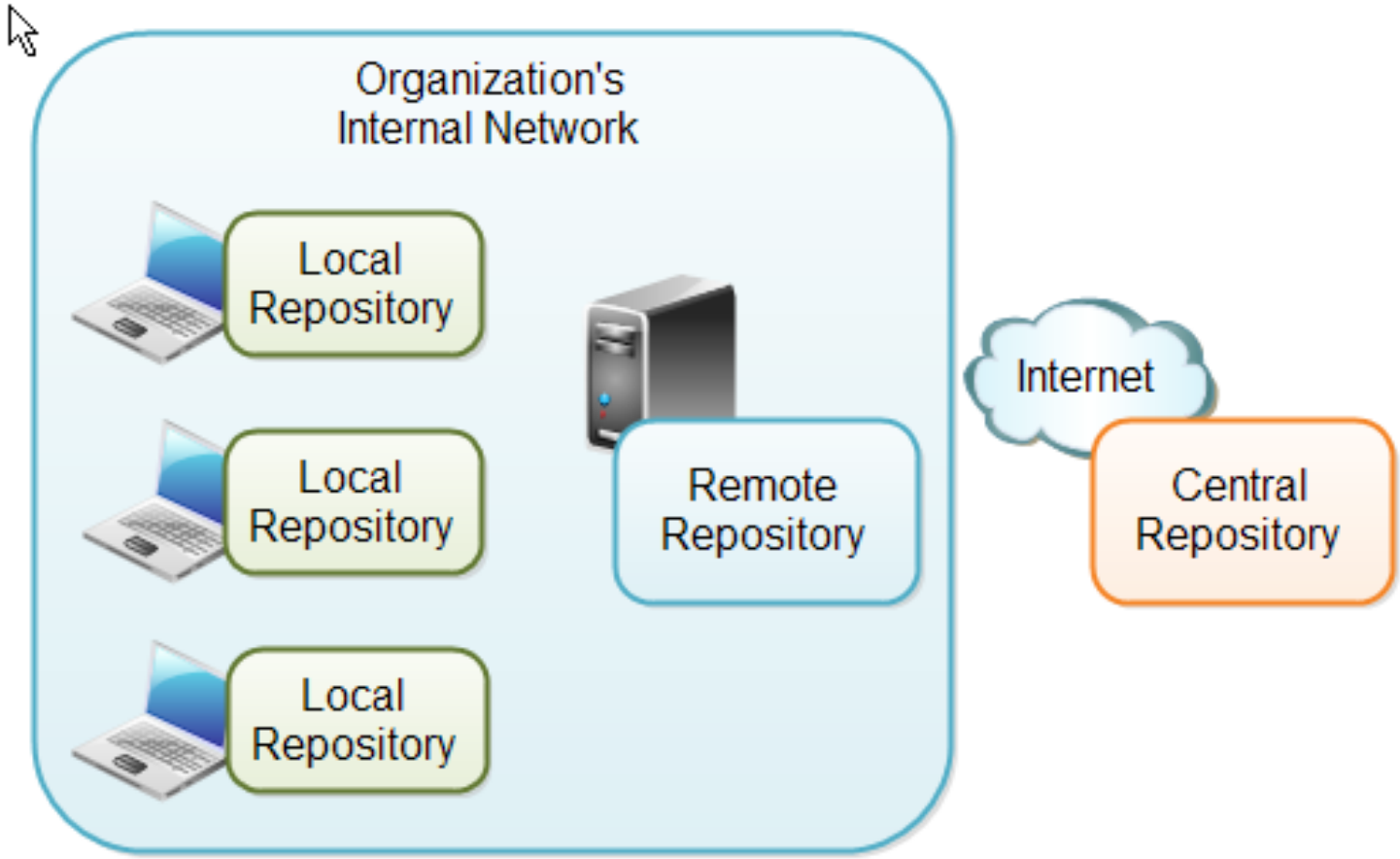
- **integration-test**: process and deploy the package if necessary into an environment where integration tests can be run
- **verify**: run any checks to verify the package is valid and meets quality criteria
- **install**: install the package into the local repository, for use as a dependency in other projects locally
- **deploy**: done in an integration or release environment, copies the final package to the remote repository for sharing with other developers and projects.

Maven Lifecycle

There are two other Maven lifecycles of note beyond the *default* list above. They are

- **clean**: cleans up artifacts created by prior builds
- **site**: generates site documentation for this project

Maven Repositories



Terminologies

- Archetype : Maven plugin whose task is to create a project structure as per its template.
- GroupId : This is an Id of project's group.
- ArtifactId : This is an Id of the project.
- Version : This is the version of the project.
- SNAPSHOT : special version that indicates a current development copy

Thank You

Santhosh Nimmani