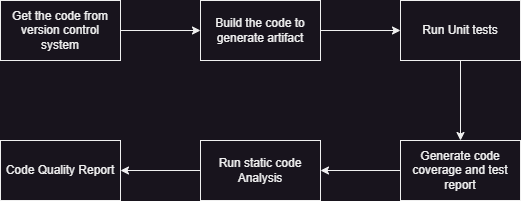
**Dependencies**

* Whenever a software project is developed, they have lot of dependencies
* before building the code, dependencies have to be present locally
* To manage these dependencies, every programming language has some kind of package manager
  + dotnet: nuget is the package manager
    - packages.config/packages.json
  + java: maven can handle package management
  + python: pip is the package manager
  + nodejs: npm can handle package management
* Scope of Work  
  

**Maven**

* Maven is a tool which can be used to build, package, distribute, test and generate documentation for java and java based languages
* Maven follows convention over configuration
* The maven uses a file called as pom.xml
* pom (Project object model)
* Maven Installation:

# install java 17

sudo apt update

sudo apt install openjdk-17-jdk -y

Let’s try installing maven 3.9.3 <https://maven.apache.org/download.cgi>

cd /tmp

wget https://dlcdn.apache.org/maven/maven-3/3.9.3/binaries/apache-maven-3.9.3-bin.tar.gz

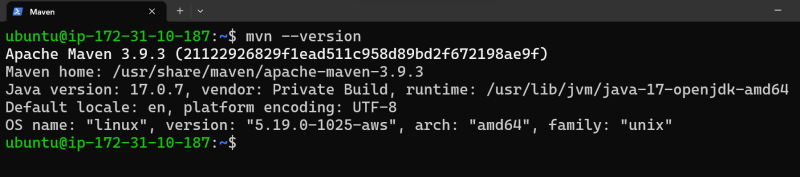
sudo mkdir /usr/share/maven

sudo tar -xvzf apache-maven-3.9.3-bin.tar.gz -C /usr/share/maven

# add /usr/share/maven/apache-maven-3.9.3/bin to the PATH variable

# add to ~/.bashrc or /etc/environment

mvn --version

  
\* Maven goals <https://www.baeldung.com/maven-goals-phases>   
\* validate: validates the pom and its project  
\* compile: this converts the java code into byte code (.java to .class). It stores the class files in target/classes  
\* test: it will run the unit tests written and generates test results in xml format in text format. folder will be /target/surefire-reports/TEST-\*.xml  
\* package: This creates the packaging format (jar/war/ear). will be <artifact-id>-<version>.<packaging-format>  
\* install: This copies the package and its definition into M2\_HOME or ~/.m2/repository  
\* deploy: copying package and its definition to remote repository for other users in other systems to use what you have built  
\* clean: clean removes target folder  
\* Lifecycle <https://maven.apache.org/guides/introduction/introduction-to-the-lifecycle.html#Lifecycle_Reference>   
\* To execute any lifecycle goal mvn <goal>  
\* We have written a simple pom

<project>

<modelVersion>4.0.0</modelVersion>

<groupId>io.learningthoughts.samples</groupId>

<artifactId>hello-maven</artifactId>

<version>1.0.0-SNAPSHOT</version>

<properties>

<maven.compiler.source>11</maven.compiler.source>

<maven.compiler.target>11</maven.compiler.target>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>

* Maven packaging formats: <https://www.baeldung.com/maven-packaging-types#:~:text=Maven%20offers%20many%20default%20packaging%20types%20that%20include%20a%20jar,and%20performs%20a%20specific%20task>.
* Maven goals downloads dependencies and stores in
  + M2\_HOME where ever this environment variable points to and if not found does in <home-dir>/.m2

**Terms**

* Artifact => For generating artifacts we use build tools msbuild, maven/gradle
* unit test => junit, mstest/nunit, pytest, jasmine, mocha, most ci cd systems understand junit xml reports to generate test results
* code coverage => we do this from sonar qube
* Static Code Analysis => we do this from sonar qube
* artifact repository => we would use jfrog (azure artifacts)