Build Lifecycle  
Demonstrate the use of Maven

**Build Lifecycle**

The build lifecycle is a series of phases that are executed in a specific order to build, test, and deploy a software project. The build lifecycle typically includes the following phases:

* Validate: Validate the project's dependencies and configuration.
* Compile: Compile the project's source code into bytecode.
* Test: Run unit tests to ensure the code is working as expected.
* Package: Package the compiled code into a distributable format (e.g., JAR, WAR).
* Verify: Verify the packaged artifact meets the project's quality standards.
* Install: Install the packaged artifact into a local repository.
* Deploy: Deploy the packaged artifact to a remote repository or production environment.

**Maven Demonstration**

Maven is a popular build tool that automates the build lifecycle process. Here's a step-by-step demonstration of how to use Maven:

**Step 1: Create a new Maven project**

Open a terminal or command prompt and run the following command to create a new Maven project:

mvn archetype:generate -DgroupId=com.example -DartifactId=myproject -DarchetypeArtifactId=maven-archetype-quickstart

This will create a new directory called myproject with a basic Maven project structure.

**Step 2: Configure the project**

Open the pom.xml file in the myproject directory and add the following dependencies:

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.12</version>

<scope>test</scope>

</dependency>

</dependencies>

This adds the JUnit dependency for unit testing.

**Step 3: Write some code**

Create a new Java class in the src/main/java directory:

public class Hello {

public static void main(String[] args) {

System.out.println("Hello, World!");

}

}

**Step 4: Compile the code**

Run the following command to compile the code:

Mvn compile

This will compile the Java code into bytecode.

**Step 5: Run unit tests**

Create a new test class in the src/test/java directory:

public class HelloTest {

@Test

public void testHello() {

Hello hello = new Hello();

assertEquals("Hello, World!", hello.getMessage());

}

}

Run the following command to run the unit tests:

mvn test

This will execute the unit tests and report any failures.

**Step 6: Package the code**

Run the following command to package the code into a JAR file:

mvn package

This will create a myproject-1.0-SNAPSHOT.jar file in the target directory.

**Step 7: Install the package**

Run the following command to install the package into the local repository:

mvn install

This will install the package into the local Maven repository.

**Step 8: Deploy the package**

Run the following command to deploy the package to a remote repository:

mvn deploy

This will deploy the package to a remote repository, such as Maven Central.