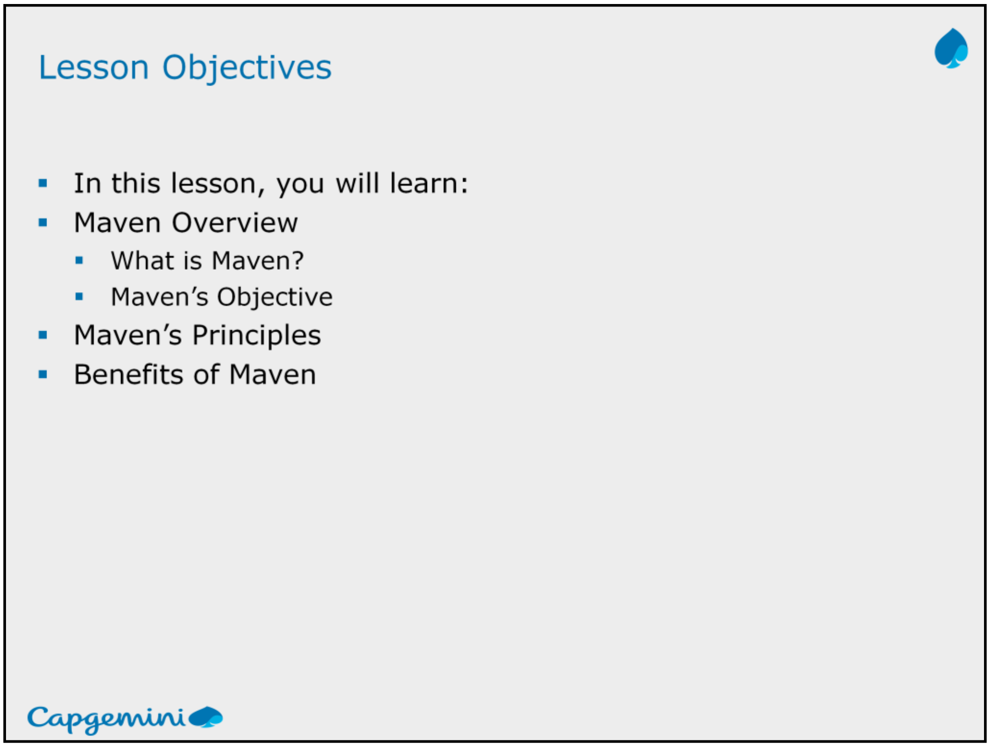
Instructor Notes:

Maven

- Introduction to Maven

Instructor Notes:



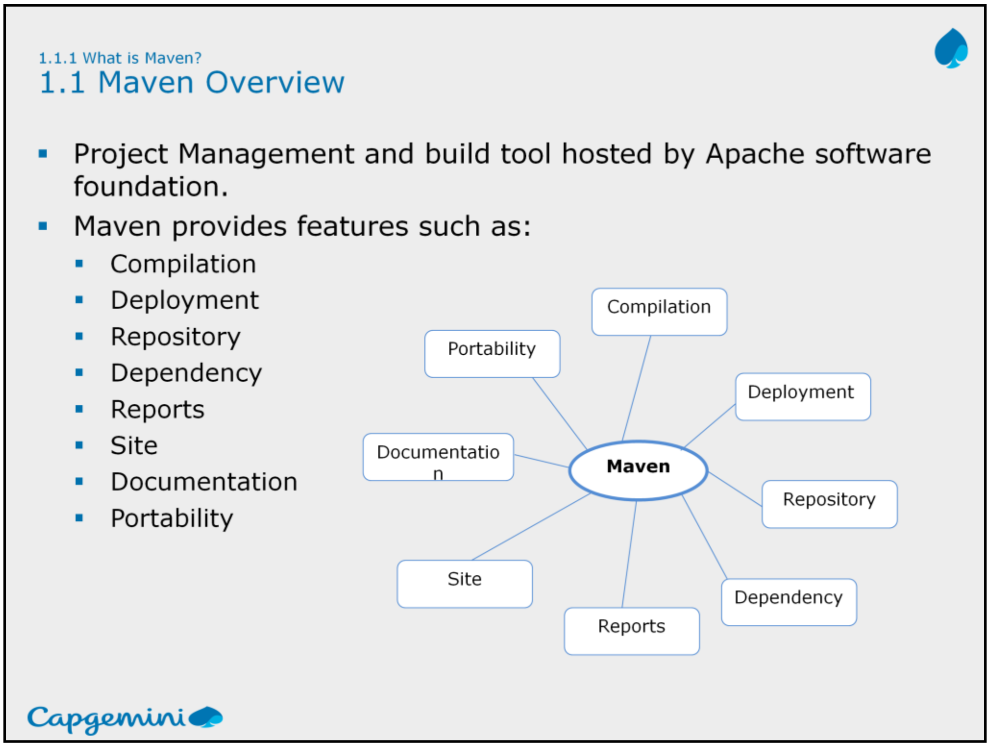
Lesson Objectives

- In this lesson, you will learn:
- Maven Overview
  - What is Maven?
  - Maven’s Objective
- Maven’s Principles
- Benefits of Maven

Instructor Notes:

Ant is Task based build tool. Maven brings Object Oriented approach to build table.

A build tool such as Ant is focused solely on preprocessing, compilation, packaging, testing, and distribution. A project management tool such as Maven provides a superset of features found in a build tool. In addition to providing build capabilities, Maven can also run reports, generate a web site, and facilitate communication among members of a working team.



1.1.1 What is Maven?

1.1 Maven Overview

- Project Management and build tool hosted by Apache software foundation.
- Maven provides features such as:
  - Compilation
  - Deployment
  - Repository
  - Dependency
  - Reports
  - Site
  - Documentation
  - Portability

Compilation
Portability
Deployment
Documentatio
n
Maven
Repository
Site
Dependency
Reports
Capgemini

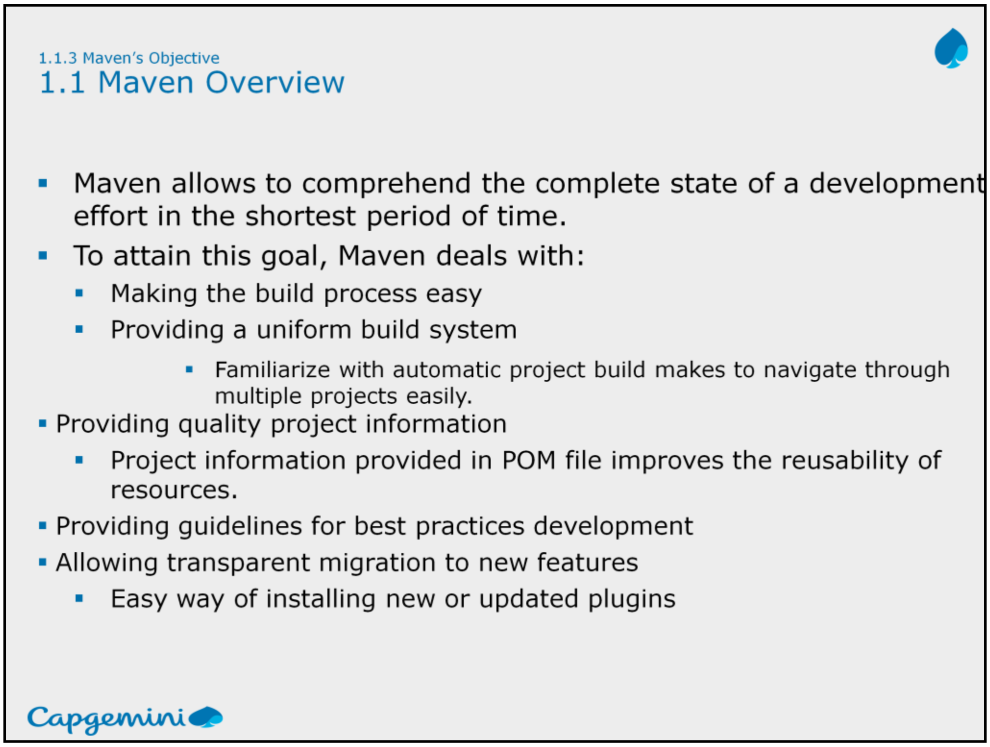
- **Maven** is a tool for project management and build automation.
- **Maven is not Ant++**
- Maven serves a similar purpose to the Apache Ant tool, but it is based on different concepts and works in a profoundly different manner.
- Maven is hosted by the Apache Software Foundation, where it was formerly part of the Jakarta Project**.**
- It is pronounced as “May-ven” and Maven is a Yiddish (Jewish Lang) word which means “**accumulator of knowledge**“.

Actually, there is a open source project named Jakarta Turbine which was facing complexity issues with module build processes and atlast the Apache team came up with a solution to simplify this build process which was than termed as Apache Maven.

Maven provides features such as:

- Build tool Capabilities
- Run Reports
- Generate a website
- Dependency Management
- Repositories (Reusable Plug-ins)
- Continuous Integration build systems
- Portable
- Building configuration using maven are portable to another machine without any effort

Instructor Notes:



1.1.3 Maven's Objective

1.1 Maven Overview

- Maven allows to comprehend the complete state of a development effort in the shortest period of time.
- To attain this goal, Maven deals with:
  - Making the build process easy
  - Providing a uniform build system
      - Familiarize with automatic project build makes to navigate through multiple projects easily.
- Providing quality project information
  - Project information provided in POM file improves the reusability of resources.
- Providing guidelines for best practices development
- Allowing transparent migration to new features
  - Easy way of installing new or updated plugins

Capgemini

Maven allows to comprehend the complete state of a development effort in the shortest period of time. To attain this goal, Maven deals with:

Making the build process easy:

While using Maven doesn't eliminate the need to know about the underlying mechanisms, Maven does provide a lot of shielding from the details.

Providing a uniform build system:

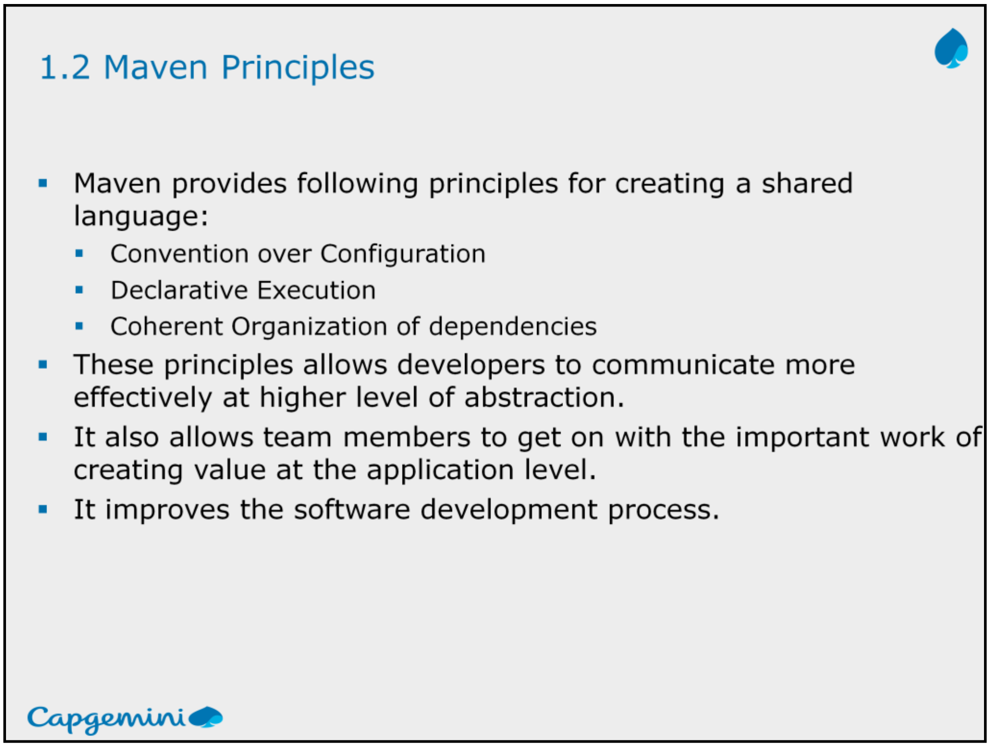
Maven allows a project to build using its project object model (POM) and a set of plugins that are shared by all projects using Maven, providing a uniform build system. Once you familiarize yourself with how one Maven project builds you automatically know how all Maven projects build saving you immense amounts of time when trying to navigate many projects.

Providing quality project information:

Maven provides plenty of useful project information that is in part taken from your POM and in part generated from your project's sources. For example, Maven can provide:

- Change log document created directly from source control
- Cross referenced sources
- Mailing lists
- Dependency list
- Unit test reports including coverage

Instructor Notes:

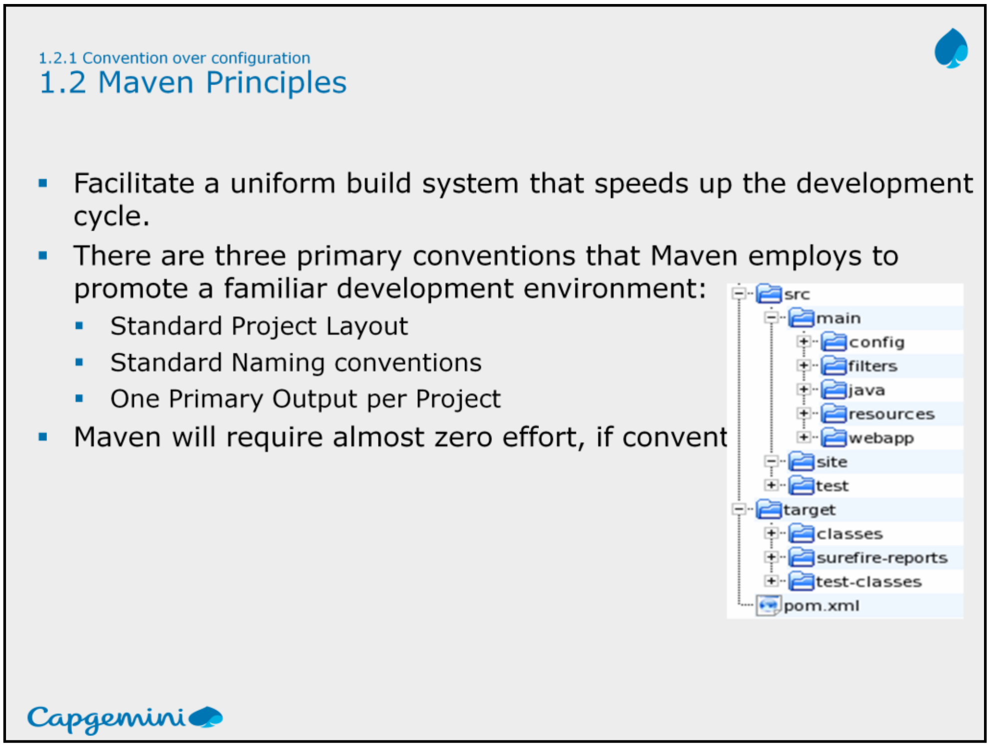


1.2 Maven Principles

- Maven provides following principles for creating a shared language:
  - Convention over Configuration
  - Declarative Execution
  - Coherent Organization of dependencies
- These principles allows developers to communicate more effectively at higher level of abstraction.
- It also allows team members to get on with the important work of creating value at the application level.
- It improves the software development process.

Instructor Notes:

Zero effort – No need to specify the location of source file where it is. Just place source file in the correct directory, maven will take care of the rest while building the project.



1.2.1 Convention over configuration

1.2 Maven Principles

- Facilitate a uniform build system that speeds up the development cycle.
- There are three primary conventions that Maven employs to promote a familiar development environment:
  - Standard Project Layout
  - Standard Naming conventions
  - One Primary Output per Project
- Maven will require almost zero effort, if convent

Conventions over configuration used to facilitate a uniform build system that speeds up the development cycle.

There are three primary conventions that Maven employs to promote a standardized development environment:

Standard Project Layout:

Having a common directory layout would allow for users familiar with one Maven project to immediately feel at home in another Maven project. The advantages are analogous to adopting a site-wide look-and-feel.

Standard Naming conventions

If multiple projects are involved, standard naming convention for directories provide clarity and immediate comprehension.

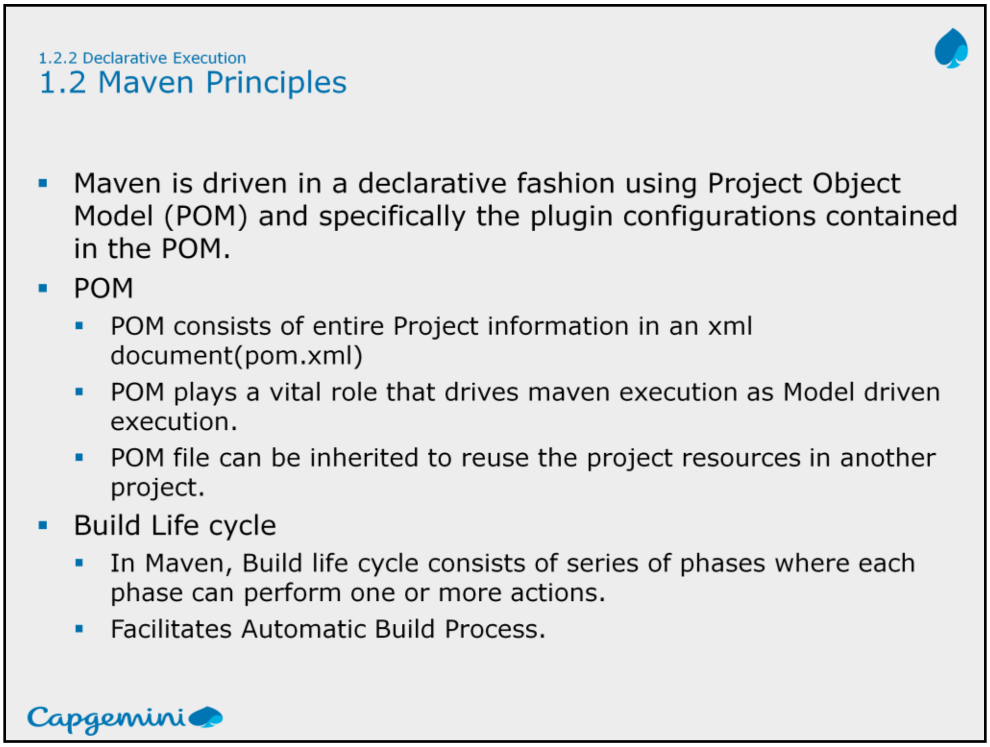
One Primary Output per Project:

Instead of producing a single JAR file for entire project, Maven would encourage you to have three, separate projects: a project for the client portion of the application, a project for the server portion of the application, and a project for the shared utility code portion.

This separation of concerns (SoC) principle used to achieve the required engineering quality factors such as adaptability, maintainability, extendibility and reusability.

If you follow the convention, Maven will require almost zero effort - just put your source in the correct directory and Maven will take care of the rest.

Instructor Notes:



1.2.2 Declarative Execution

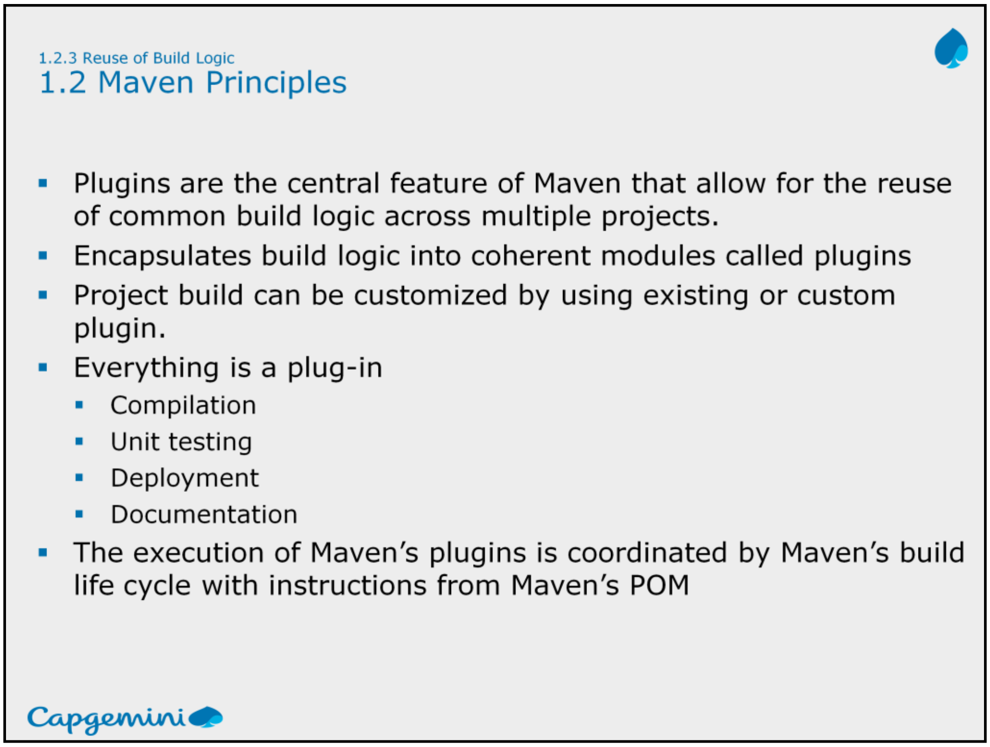
1.2 Maven Principles

- Maven is driven in a declarative fashion using Project Object Model (POM) and specifically the plugin configurations contained in the POM.
- POM
  - POM consists of entire Project information in an xml document(pom.xml)
  - POM plays a vital role that drives maven execution as Model driven execution.
  - POM file can be inherited to reuse the project resources in another project.
- Build Life cycle
  - In Maven, Build life cycle consists of series of phases where each phase can perform one or more actions.
  - Facilitates Automatic Build Process.

Capgemini

Maven is driven in a declarative fashion using Maven's Project Object Model (POM) and specifically the plug-in configurations contained in the POM.

Instructor Notes:



1.2.3 Reuse of Build Logic

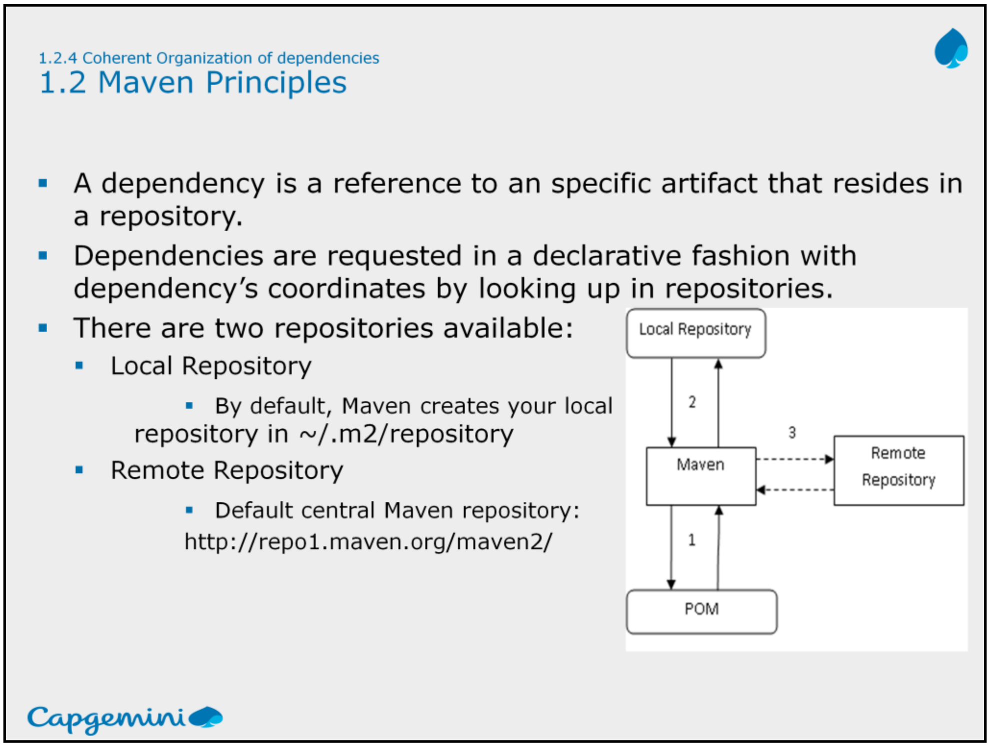
1.2 Maven Principles

- Plugins are the central feature of Maven that allow for the reuse of common build logic across multiple projects.
- Encapsulates build logic into coherent modules called plugins
- Project build can be customized by using existing or custom plugin.
- Everything is a plug-in
  - Compilation
  - Unit testing
  - Deployment
  - Documentation
- The execution of Maven’s plugins is coordinated by Maven’s build life cycle with instructions from Maven’s POM

Capgemini

Maven uses SoC principle to promote reuse of build logic by encapsulating build logic into coherent modules called plugins. In Maven, there is a plug-in for compiling source code, a plug-in for running tests, a plug-in for creating JARs, a plug-in for creating Javadocs, and many other functions. Plugins are the key building blocks for everything in Maven.

Instructor Notes:



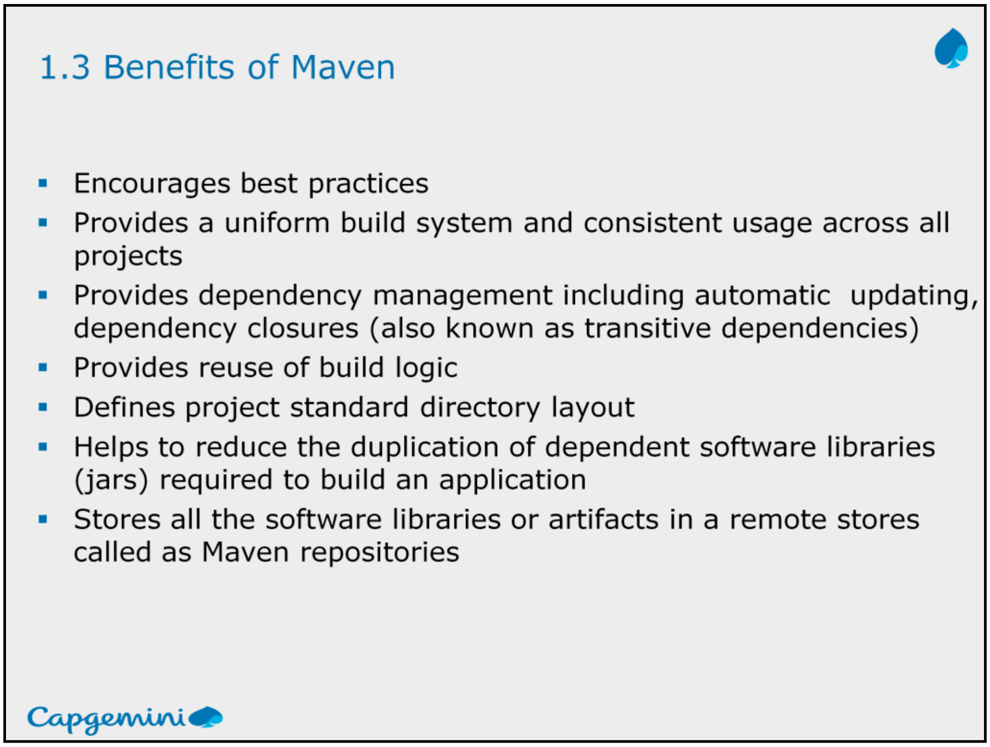
1.2.4 Coherent Organization of dependencies

1.2 Maven Principles

- A dependency is a reference to an specific artifact that resides in a repository.
- Dependencies are requested in a declarative fashion with dependency's coordinates by looking up in repositories.
- There are two repositories available:
  - Local Repository
    - By default, Maven creates your local repository in ~/.m2/repository
  - Remote Repository
    - Default central Maven repository: http://repo1.maven.org/maven2/

When a dependency is declared within the context of your project, Maven tries to satisfy that dependency by looking in repositories. Maven uses a local repository to resolve its dependencies. If not found one or more remote repositories are consulted to find a dependency. If found, the dependency is downloaded to the local repository and used from the local repository.

Instructor Notes:



1.3 Benefits of Maven

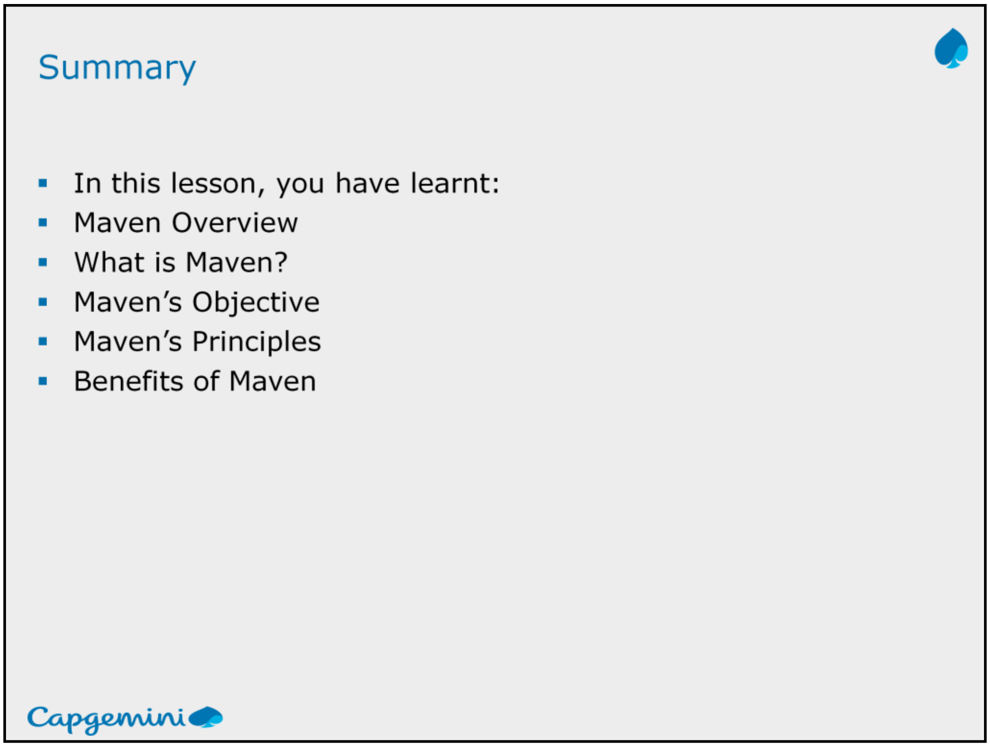
- Encourages best practices
- Provides a uniform build system and consistent usage across all projects
- Provides dependency management including automatic updating, dependency closures (also known as transitive dependencies)
- Provides reuse of build logic
- Defines project standard directory layout
- Helps to reduce the duplication of dependent software libraries (jars) required to build an application
- Stores all the software libraries or artifacts in a remote stores called as Maven repositories

Capgemini

Maven can provide benefits for your build process by employing standard conventions and practices to accelerate your development cycle while at the same time helping you achieve a higher rate of success.

Instructor Notes:

Additional notes for instructor

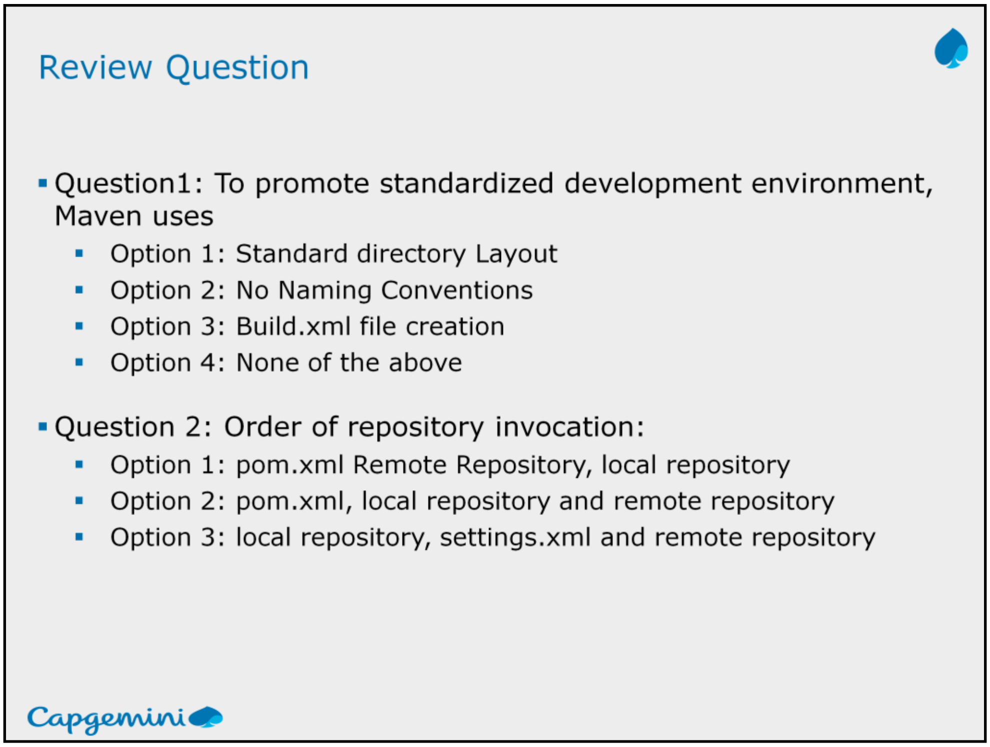


Summary

- In this lesson, you have learnt:
- Maven Overview
- What is Maven?
- Maven’s Objective
- Maven’s Principles
- Benefits of Maven

Instructor Notes:

Answers for the Review Questions:
Question 1: Option 1 and 2
Question 2: option 2



Review Question

- Question1: To promote standardized development environment, Maven uses
  - Option 1: Standard directory Layout
  - Option 2: No Naming Conventions
  - Option 3: Build.xml file creation
  - Option 4: None of the above
- Question 2: Order of repository invocation:
  - Option 1: pom.xml Remote Repository, local repository
  - Option 2: pom.xml, local repository and remote repository
  - Option 3: local repository, settings.xml and remote repository

Capgemini