BUILD OVERVIEW

Software Development Life Cycle Phases :

* Planning
* Requirements Analysis
* Design
* Implementation
* Testing
* Deployment
* Maintainance

Build Tools :

Key Functions :

* Compiling Source Code
* Packaging Binaries
* Running Tests
* Deploying Applications

Benefits :

* Automation
* Dependency Management
* Consistency

Environments :

* Development
* Staging
* Production

Build Tool Process in Software Development :

1. Identify Need for automation
2. Select Appropriate Build Tool
3. Automate Code Conversion
4. Streamline Development Workflow
5. Enhance Productivity
6. Maintain Consistency
7. Ensure Quality

Code Build :

The build process converts raw source code into an executable program that a computer can run.It is the technical compilation and linking phase that focuses on creating functional, low-level program files.

This process includes:

* Compiling source code files into machine code or bytecode.
* Linking the compiled code with other pieces of code and libraries to create a single executable file.
* Checking for code errors and verifying that all dependencies function correctly.

Packaging :

The packaging process bundles the built program with all other necessary components into a single, distributable artifact. The goal is to create a complete, ready-to-install bundle that includes everything required to run the software in a target environment.

This includes:

* The compiled code from the build process.
* Configuration files (.yml, .json)
* Required libraries and scripts
* Documentation files
* Any other assets vital for the software's operation.

Build and package the code :

The flow from development to deployment typically follows this sequence:

* Code: A developer writes source code.
* Build: A build tool compiles the code and runs tests.
* Package: The package manager creates the distributable artifact.
* Deploy: The package is moved to a production environme

PACKAGE

DEV

RELEASE

BUILD

USERS

QA

VERSION CONTROL

PROD

DEVELOPER

Web Based Application:

Create Package Suitable for the server deployment

Code and compiled

User Accessing the application

Developer writing code

User Package and deploy the server

Let us taken an example for build overview

Github

Version Control

Java, Gradle, Maven

Developer

Ubuntu Server

Java

Manual Build Process :

A manual build process refers to the steps and procedures undertaken by a human developer to compile, link, and package software without using an automated tool.

Challenges in Manual Build Process :

* Greater risk of mistakes.
* Inconsistent results
* Time-consuming
* Poor traceability

Compiler :

A compiler is a specialized computer program that translates human-readable source code written in a high-level programming language into a lower-level language, such as machine code or assembly language, that a computer's processor can directly execute. The entire program is translated before it is run.

compile

Java .class

Build Tool :

A build tool is a program that automates the process of converting source code into a final, runnable software product, such as an executable file, library, or web application. It orchestrates and manages the entire build process, which includes tasks like compiling, managing dependencies, running tests, and packaging.

Workflow for Compilation and Packaging :

1. Compile the code
2. Run Unit-Tests
3. Run Static-code analysis
4. Create a package
5. Deploy the package into test environments for testing.
6. Once testing approves it deploy to UAT/Pre-production/Staging.
7. Deploy to production

Why do we use build tools or build automation?

* In smaller Projects developers will often manually invoke the build process.
* This is not practical for larger projects.
* Where it is very hard to keep track of what needs to be built, in what sequence and what dependencies there are in the building process.
* Using an automation tool allows the build process to be more consistent.

Unit Testing :

Unit testing is a software development practice that involves testing the smallest, most isolated and testable parts of an application.

Unit testing can be broken down into two main types:

Manual unit testing: This involves a developer manually writing and executing test cases for a single unit of code and verifying the output.

Automated unit testing: Automated testing uses a framework or tool to create, run, and report on test cases.

Static-Code Analysis :

Static analysis, also known as Static Application Security Testing (SAST), is a "white-box" approach that reviews the application's source code, bytecode, or binary code to find defects.

Ex:Sonarqube

Dynamic Code Analysis :

Dynamic analysis, also called Dynamic Application Security Testing (DAST), is a "black-box" approach that evaluates the software's behavior by executing it in a real or simulated environment.

Popular Build Tools :

* Java : Maven, Gradle, Ant
* Dot Net : MS Build
* Node JS : Gulp, Grunt
* C/C++ : Make, cmake, bmake etc.

Which Build tool should I choose for my Project?

* Apache maven : Maven is a popular project management and build tool that uses a declarative, "convention-over-configuration" approach.
* Gradle : Gradle is a more modern, flexible, and high-performance build automation system that builds upon the concepts of Ant and Maven.
* Ant : Ant is the oldest of the three build tools and is highly flexible but procedural.