**Session 15**

**Assignment 2**

**Problem Statement:**

Explain the working and the differences between Maven, Gradle and SBT in detail.

**Part 1:** Explain the working of Maven, Gradle and SBT in detail.

**Maven:**

Maven is a "build management tool", it is for defining how your .java files get compiled to .class, packaged into .jar (or .war or .ear) files, (pre/post)processed with tools, managing your CLASSPATH, and all others sorts of tasks that are required to build your project.

Maven’s primary goal is to allow a developer to comprehend the complete state of a development effort in the shortest period of time. In order to attain this goal there are several areas of concern that Maven attempts to deal with:

* Making the build process easy
* Providing a uniform build system
* Providing quality project information
* Providing guidelines for best practices development
* Allowing transparent migration to new features

It is similar to Apache Ant or Gradle or Makefiles in C/C++.

It is also designed around the "build portability" theme, so that you don't get issues as having the same code with the same buildscript working on one computer but not on another one.

**Gradle:**

Gradle is a build system and supports build-by convention.

It takes the best features from other build systems and combines them into one. It is improved based off of their shortcomings.

It is a JVM based build system as the script can be written in Java, which Android Studio makes use of.

Google found Gradle as one of the most advanced build systems on the market and realized that the scripts could be self-written by the user with little to no learning curve, and without learning Groovy or any other new language. So Google wrote the Android plugin for Gradle.

Thus the Gradle can be a plugin based system.

**Advanced features of Gradle:**

* Parallel unit test execution
* Dependency building
* Incremental build support
* Dynamic tasks and task rules
* Gradle daemon

**SBT:**

SBT is an open source build tool for Scala and Java projects, similar to Java's Maven or Ant. It is a tool to define task and then run them from the shell.

The main features of SBT are:

* Native support for compiling Scala code and integrating with many Scala test frameworks
* Build descriptions written in Scala using a DSL
* Dependency management using Ivy (which supports Maven-format repositories)
* Continuous compilation, testing, and deployment
* Integration with the Scala interpreter for rapid iteration and debugging
* Support for mixed Java/Scala projects

SBT is the ‘DE FACTO’ build tool in the Scala community.

**Part 2:** The differences between Maven, Gradle and SBT.

Comparison of Maven and Gradle on the basis of features:

* How easy is the initial learning curve

1. Maven is XML based tool, XML is very commonly used/known.
2. If existing project are using maven then developers are comfortable with the system.
3. Gradle is DSL based system and need to learn explicitly.

* How fast are different builds with each tool

1. Taken a report from zero turn around, they have done detail analysis of the speed of builds with both built tools and found maven and Gradle are close enough on build timings.

* How complex is it to create and maintain the build script?

1. Maven build scripts are xml based, which has predefined structure and only one way to write.
2. So it makes it more standard and less flexible.
3. Gradle has its own DSL which is introduced by Gradle itself and tightly connected to Gradle internals.
4. But Gradle is flexible and simple and short.

* How many plugins exist and how simple is it to customize your own plugins?

1. Maven is called “plugin execution framework”.
2. Hundreds of plugins exist for Maven and you can create your own plugin is simple.
3. Gradle’s architecture is also plugin-based.
4. It is easy to write plugins but availability of plugin in community may not be ease.

* How good is the community and documentation for each tool?

1. Maven is in the market for very long time.
2. Documentation is good.
3. Lots of resources and help available in open community and forum.
4. Gradle is very recent and it is open source but still under control of gradleware.
5. They have option for commercial support.

* How well does each tool integrate with developer tools?

1. With many years of background Maven has full support to almost each tool and every category (IDE, App Server, and CI).
2. Lacking in the App Server and CI Server Categories, mainly due to newness.