Byggverktyg

Vilka finns?
Och vad är de bra för respektive dåliga på?

ANT

Another Neat Tool

- Low learning curve
- Procedural programming idea
- Control of Build Process
- Flexible
- XML is hierarchical
- Big long boring file

Maven

- XML
- Convention over Configuration
- Dependency Management
- Strict Project Structure

Gradle

- Uses the best of both Ant and Maven
- No XML
- Smaller configuration files
- High Learning Curve
- Groovy

Utan Tester: 7.378	6.279	3.342
Med Tester: 10.705	13.357	10.030

Maven

Kräver JDK finns på datorn. 1.7 eller högre för Maven 3.3+

Bygger på Dependency Management.
Alla dependencies laddas ner från ett Central Repository.

Om man inte specificierar en version, laddas senaste versionen ner.

En stor nackdel är att konflikter kommer uppstå. Olika versioner av samma bibliotek kan skapa fel, som lättast löses med att man tar bort alla dependencies från utvecklingsmiljön och laddar hem dem igen.

En annan nackdel är att pom filen kan bli extremt stor och oöverskådlig. Man kan bryta upp i en "parent" pom.

Likt Java som alltid ärver klassen Object, så varje pom fil ärver The SUPER-POM

All konfigurering hanteras i en specifik fil: pom.xml

Filen har speciella taggar som förklarar för Maven om det är en Dependency, Plugin, Build, Execution eller något annat.

När man kör Maven använder man speciella lifecycle hooks beroende på vad man vill göra.

På kommandoraden använder man mvn <hookname>

Lifecycle Hooks

validate – Projektet är korrekt och all information finns

compile – Kompilerar källkoden

test – Kör alla testklasser

package – Binder ihop all källkod till en JAR, WAR eller EAR fil

verify – Kontrollerar resultatet av integrationstester

install – Installerar paketet i det lokala repositoriet

deploy – Kopierar slutgiltiga produkten och skickar upp det på Central Repo

Man kan skriva flera lifecycle hooks på samma kommando, ett whitespace mellan varje hook är allt som behövs.

T:ex: mvn verify clean package

Clean – lifecycle hook? Det är en lifecycle reference för: pre-clean clean post-clean.

Den tar bort target katalogen och städar upp eventuella referenser.

Installera Maven

Ladda hem senaste Maven från maven.apache.org

Packa upp och lägg (helst under Program Files)

Lägg till en Path till bin mappen i systemets miljövariabler

Testa med att starta en ny cmd, eller i Bash att skriva:
mvn --version

GRADLE

Vad är Gradle

Precis som med Maven, är Gradle ett byggverktyg.

Det kräver JVM installerat på datorn.

Gradle arbetar snabbar än Maven, och används därför by default i Android utveckling.

Man måste lära sig Groovy, vilket inte är så svårt, då man oftast skall lägga till dependencies och lägga till tasks.

Groovy har alltid fått kämpa i motvind. Det har klassificerats till att vara svårläst, jobbigt att skriva i och är ett dynamiskt språk.

Med Kotlin kom ett tillägg som gör att man kan skriva Gradle filer i Kotlin istället. Kotlin är en mix mellan OOP och funktionellt språk.

Gradle likt Maven använder sig av Build phases:

- Initialization
- Configuration
- Execution

Även om både Maven och Gradle har Buod Phases skiljer dem sig åt, då Gradles Phases är mer flexibla. Man kan skriva om dem i sin *settings.gradle*.