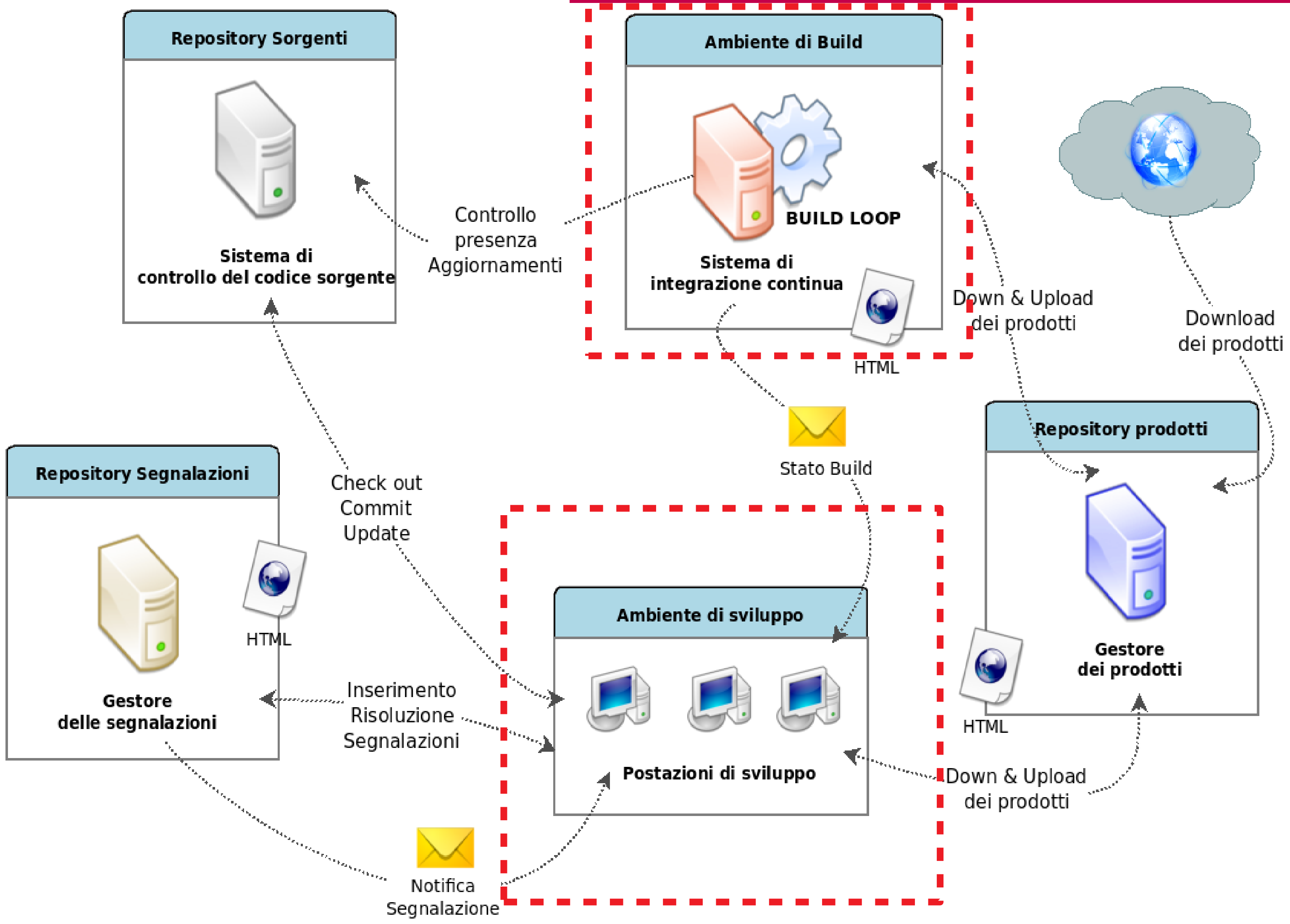
# METODI E TECNOLOGIE PER LO SVILUPPO SOFTWARE

Nicola Bertazzo
nicola.bertazzo [at] unipd.it

Università degli Studi di Padova
Dipartimento di Matematica
Corso di Laurea in Informatica, A.A. 2022 – 2023

## Visione Generale



## Definizione

**Build automation** is the process of **automating** the creation of a **software build** and the associated processes including: compiling computer source code into binary code, packaging binary code, and running automated tests.

Historically, build automation was accomplished through makefiles. Today, there are two general categories of tools:

- **Build-automation utility** (like Make, Rake, Cake, MS build, Ant, Gradle etc.)
  Whose **primary purpose is to generate build artifacts** through activities like compiling and linking source code.

- **Build-automation servers**
  These are general web based tools that **execute build-automation utilities on a scheduled or triggered basis**; a continuous integration server is a type of build-automation server.

WIKIPEDIA
The Free Encyclopedia

## Definizione

**Build automation** is the process of **automating** the creation of a **software build** and the associated processes including: compiling computer source code into binary code, packaging binary code, and running automated tests.

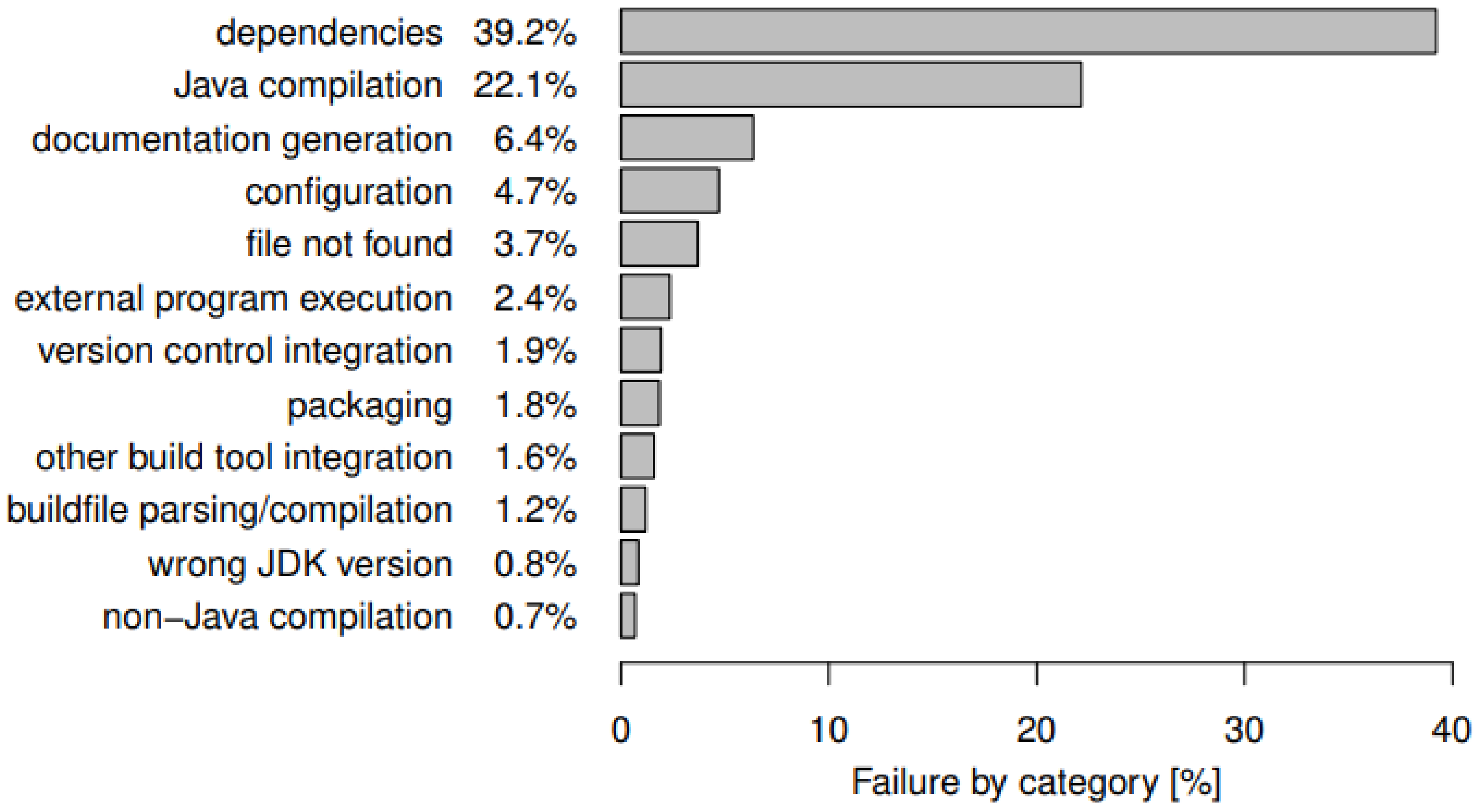
Historically, build automation was accomplished through makefiles. Today, there are two general categories of tools:

- **Build-automation utility** (like Make, Rake, Cake, MS build, Ant, Gradle etc.)
Whose **primary purpose is to generate build artifacts** through activities like compiling and linking source code.

- **Build-automation servers**
These are general web based tools that **execute build-automation utilities on a scheduled or triggered basis**; a continuous integration server is a type of build-automation server.

WIKIPEDIA
The Free Encyclopedia

## Java Projects Build Errors



https://sulir.github.io/papers/Sulir16quantitative.pdf

## Integrazione

“ In my early days in the software industry, one of the **most awkward and tense moments of a software project was integration**. Modules that **worked individually** were **put together** and the whole **usually failed** in ways that were infuriatingly difficult to find. Yet in the last few years, integration has largely vanished as a source of pain for projects, diminishing to a non-event. “
https://martinfowler.com/books/duvall.html

IEEE Software paper in 2003, 27% companies surveyed run daily builds.
State of Agile, 2013, 69% **automated build**, 57% use continuous integration tools.

## Esigenze

Poter scaricare, compilare ed utilizzare un progetto open-source

Collaborare ad un progetto e poter provare le modifiche effettuate

Modificare un bug presente in una dipendenza (p.es. Libreria)

**Riassumendo:**

A partire dal codice sorgente, creare un artefatto utilizzabile

## Tipi di build tools (Build-automation utility)

**Scripting** tools**:**
Tramite l’utilizzo di un linguaggio di scripting si definisce un automatismo (tra cui il processo di build (ad hoc)).

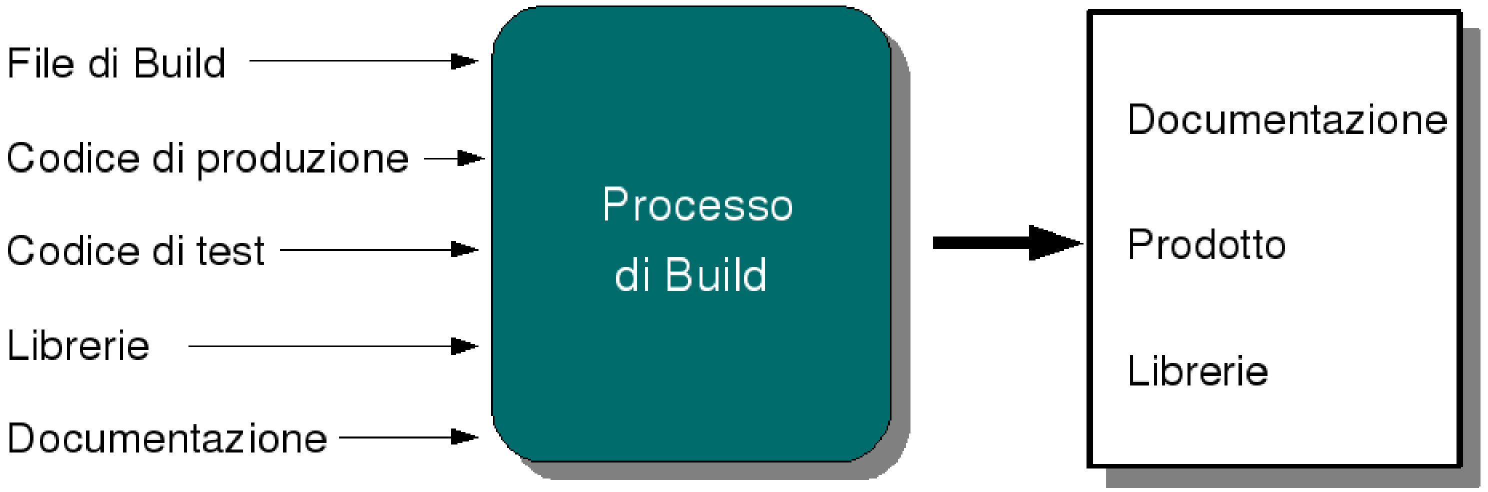
- Script .sh, .bat
- Make e Makefile: creato da Stuart Feldman nel 1977 nei laboratori Bell (Bell Labs)
- Apache Ant
- Gradle
- Rake
- Grunt

**Artifact oriented** tools**:**
Lo strumento si basa sulla definizione e creazione di un artefatto (prodotto). Viene configurato il processo build che è definito nello strumento.

- **Apache Maven**
- NPM

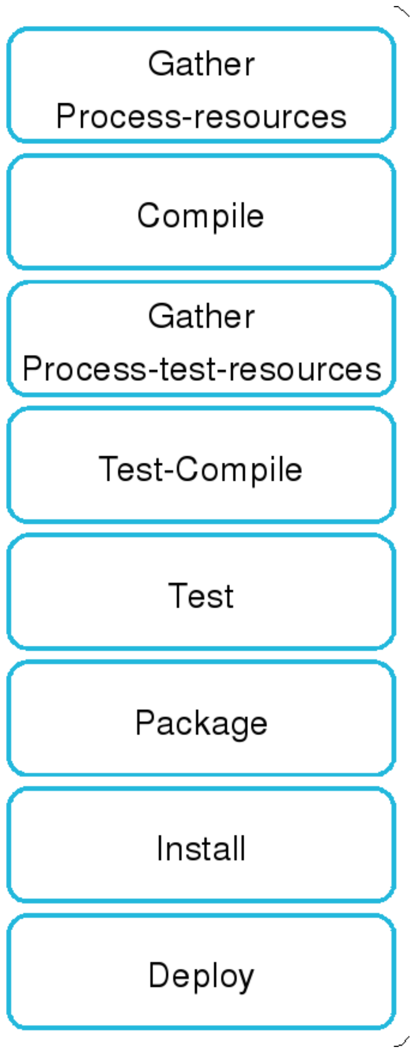
## Processo di Build

Il **processo di build** è un insieme di passi che trasformano gli script di build, il codice sorgente, i file di configurazione, la documentazione e i test in un prodotto software distribuibile



## Caratteristiche (CRISP)

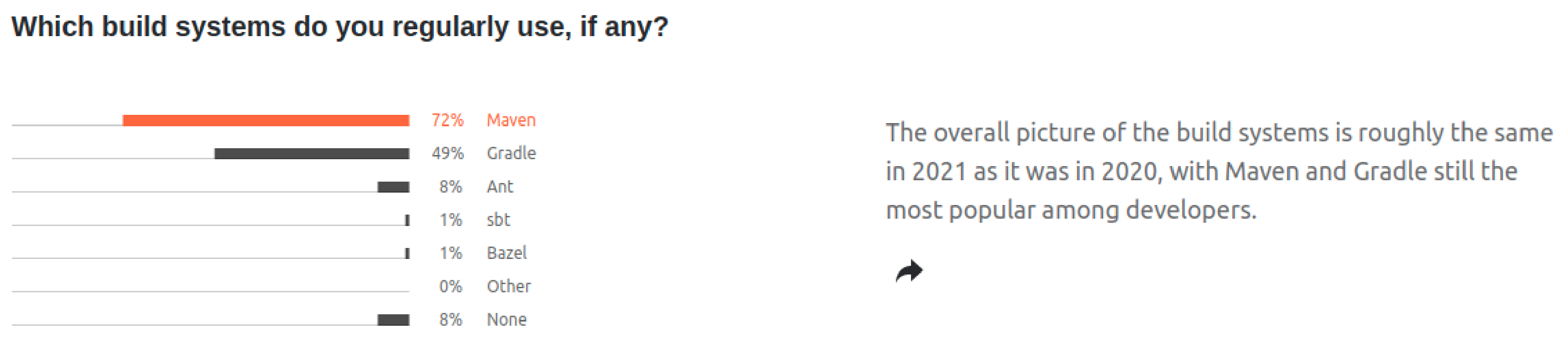
- **Completo**: Indipendente da fonti non specificate nello script di build
- **Ripetibile**: Accede ai file contenuti nel sistema di gestione del codice sorgente. Una esecuzione ripetuta dà lo stesso risultato
- **Informativo**: Fornisce informazioni sullo stato del prodotto
- **Schedulabile**: Può essere programmato ad una certa ora e fatto eseguire automaticamente
- **Portabile**: Indipendente il più possibile dall'ambiente di esecuzione



## Definizione

**Apache Maven** is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a **project's build**, **reporting** and **documentation** from a central piece of information.

Sfrutta il paradigma ***"Convention over configuration"*** che prevede una configurazione minima (o addirittura assente) per il programmatore che utilizza un framework (come maven) che lo rispetti, obbligandolo a configurare solo **gli aspetti che si differenziano** dalle **implementazioni standard** o che non rispettano particolari convenzioni di denominazione o simili.



**Which build systems do you regularly use, if any?**

| | |
|---|---|
| 72% | Maven |
| 49% | Gradle |
| 8% | Ant |
| 1% | sbt |
| 1% | Bazel |
| 0% | Other |
| 8% | None |

The overall picture of the build systems is roughly the same in 2021 as it was in 2020, with Maven and Gradle still the most popular among developers.

https://www.jetbrains.com/lp/devecosystem-2021/java/

## Caratteristiche

Acune delle principali cartatteristiche di maven sono:

**Build Tool:** sono definite delle Build Lifecycle che permettono di configurare ed eseguire il processo di build (e altri processi)

**Dependency Management:** Le dipendenze di progetto vengono specificate nel file di configurazione (pom.xml). Maven si occupa di scaricarle in automatico da dei Repository remoti e salvarle in un repository locale.

**Remote Repositories:** sono stati definiti dei repository remoti dove sono presenti gran parte delle librerie di progetti opensource e dei plugin utilizzati da maven per implementare e estendere le fasi dei Build Lifecycle

**Universal Reuse of Build Logic:** Le Build Lifecycle, i plugin maven permettono di definire in modo riusabile i principali aspetti richiesti per la gestione di progetto. Tra cui: l'esecuzione del processo di build, l'esecuzione di framework di test (p.es. Junit/TestNG), la creazione di template di progetto (p.es. Applicazioni Java Web o applicazioni costriuite con un determinato framework)

## Build Lifecycle

Maven is **based around the central concept of a build lifecycle**. What this means is that **the process for building and distributing a particular artifact (project) is clearly defined**.

For the **person building a project**, this means that it **is only necessary to learn a small set of commands to build** any Maven project, and the POM will ensure they get the results they desired.

There are three built-in build lifecycles:

- The **default** lifecycle handles your project deployment
- the **clean** lifecycle handles project cleaning
- site **lifecycle** handles the creation of your project's site documentation.

## Build Lifecycle (default)

La ***default*** Build Lifecycle è composta dalle seguenti fasi (o goals):

- **validate** - validate the project is correct and all necessary information is available
- **compile** - compile the source code of the project
- **test** - test the compiled source code using a suitable unit testing framework. These tests should not require the code be packaged or deployed
- **package** - take the compiled code and package it in its distributable format, such as a JAR.
- **verify** - run any checks on results of integration tests to ensure quality criteria are met
- **install** - install the package into the local repository, for use as a dependency in other projects locally
- **deploy** - done in the build environment, copies the final package to the remote repository for sharing with other developers and projects.

## Build Lifecycle (default)

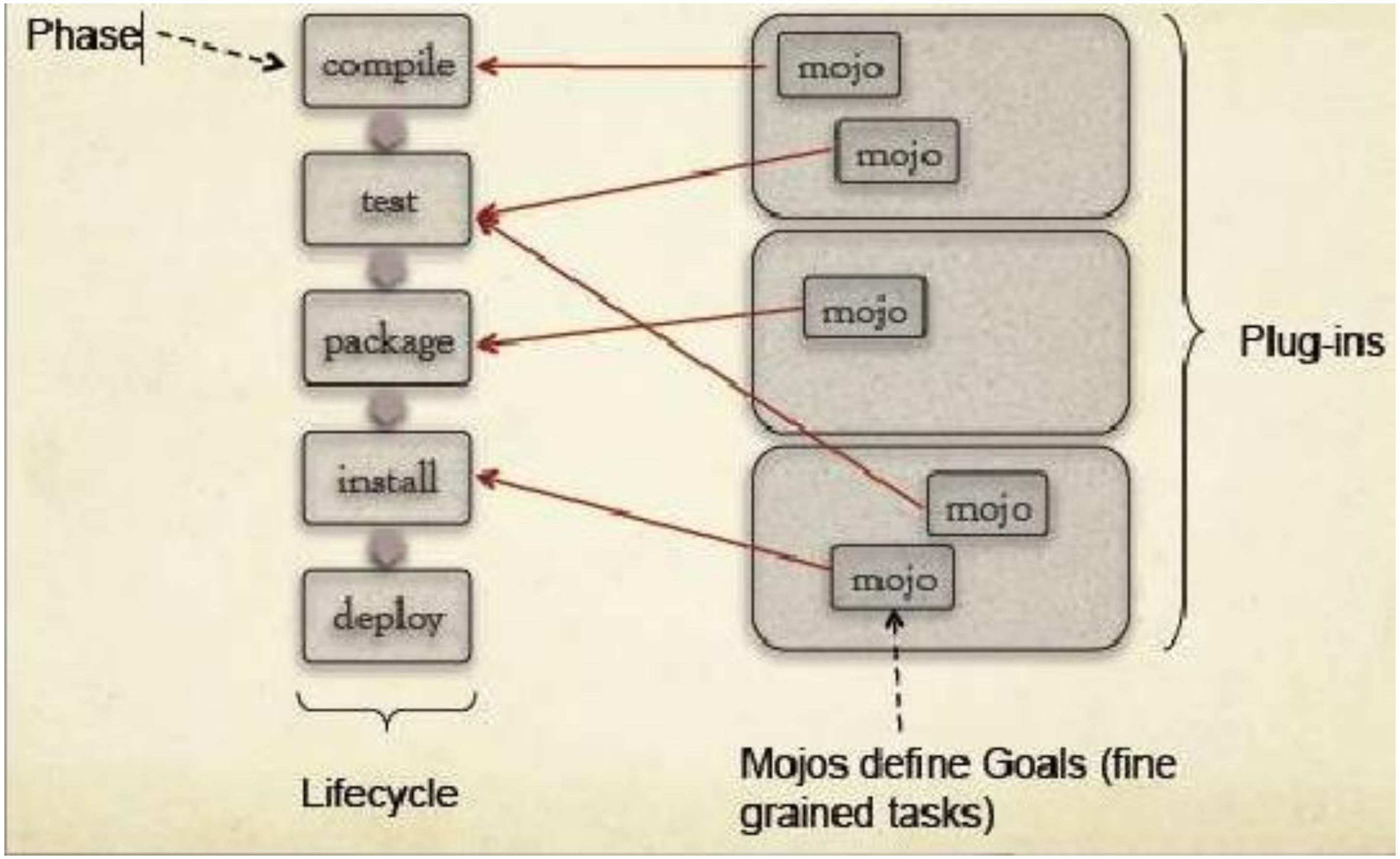
Se un progetto è gestito con maven (ed è presente il file pom.xml) sarà possibile eseguire il uno dei processi (Build Lifecycle) invocando una delle fasi predefinite.

**In un qualsiasi progetto gestito con maven**, invocando il comando:
*mvn install*

Verrà effettuata la compilazione, eseguiti i test, costruito l’artefatto e copiato nel repository locale (eseguite tutte le fasi del Build lifecycle default precedenti a install)

Le fasi vengono gestite da dei plugin maven (attraverso dei mojo)

## Build Lifecycle (default)



## POM

A **Project Object Model** or POM is the fundamental unit of work in Maven. It is an **XML file** that contains **information about the project** and **configuration details** used by Maven to build the project.

...

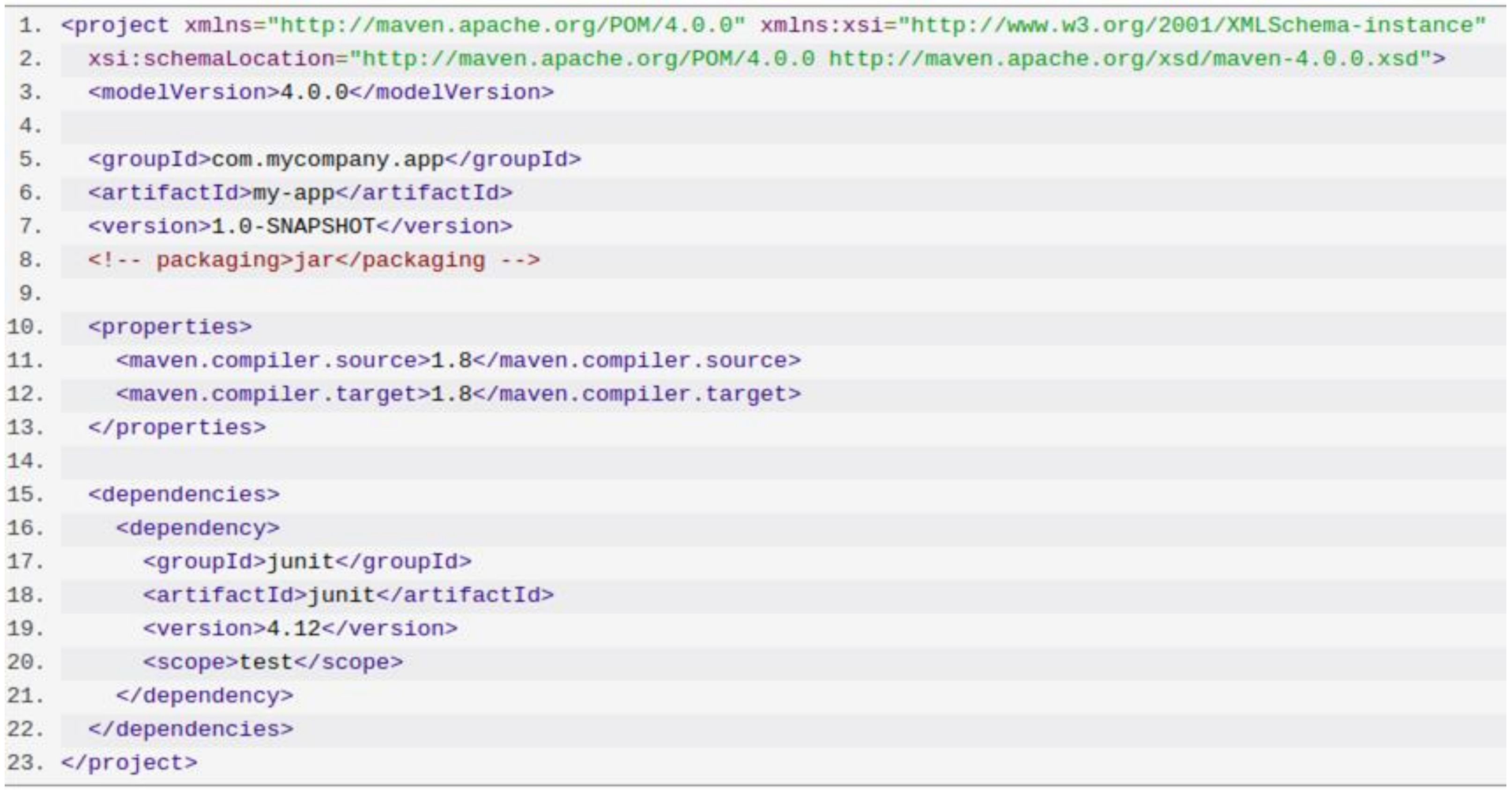
When executing a **task or goal**, Maven looks for the POM in the current directory. It reads the POM, **gets the needed configuration information**, then executes the goal.

…

Some of the configuration that can be specified in the POM are:

- the project dependencies
- the plugins or goals that can be executed
- the build profiles, and so on
- other information such as the project version, description, developers, mailing lists and such can also be specified.

## POM



```xml
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.mycompany.app</groupId>
  <artifactId>my-app</artifactId>
  <version>1.0-SNAPSHOT</version>
  <!-- packaging>jar</packaging -->

  <properties>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
  </properties>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.12</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

Questo pom è sufficiente per eseguire la build di un progetto java (e molto altro).
Le informazioni non specificate in questo file vengono ereditate da maven (convention over configuration) e sono riportate nel super POM presente nell’installazione di maven.

## Project Archetypes

In short, Archetype is a **Maven project templating toolkit**. An archetype is defined as an **original pattern or model** from which all **other things of the same kind are made**. The name fits as we are trying to provide a system that provides a consistent means of generating Maven projects. Archetype will help authors create Maven project templates for users, and provides users with the means **to generate parameterized versions of those project templates**.

mvn archetype:generate -DgroupId=com.mycompany.app -DartifactId=my-app -DarchetypeArtifactId=**maven-archetype-quickstart** -DinteractiveMode=false

Permette di creare un progetto come definito dal archetypo maven-archetype-quickstart

## Maven Plugin e Mojo

"Maven" is really just a **core framework for a collection of Maven Plugins**. In other words, **plugins are where much of the real action is performed**, plugins are used to: create jar files, create war files, compile code, unit test code, create project documentation, and on and on. **Almost any action that you can think of performing on a project is implemented as a Maven plugin.**

A Mojo is really just a goal in Maven, and plug-ins consist of any number of goals (Mojos). Mojos can be defined as annotated Java classes or Beanshell script. A Mojo specifies metadata about a goal: a goal name, which phase of the lifecycle it fits into, and the parameters it is expecting.

E' possibile sviluppare dei plugin che eseguono dei goal all'interno dei Build Lyfecycle.
I plugin vengono specificati e configurati nel pom.xml
Vengono scaricati da maven e salvati nel repository locale assieme alle altre librerie

## Riferimenti

https://en.wikipedia.org/wiki/Build_automation
https://it.wikipedia.org/wiki/Make
https://martinfowler.com/books/duvall.html
https://docs.google.com/presentation/d/1PeI-RbsisPtC8tbKMgtB3IDlffLjE6obQkp-tL0Cmsw/edit#slide=id.g1bb8ecd79c_2_27
https://sulir.github.io/papers/Sulir16quantitative.pdf
https://it.wikipedia.org/wiki/Convention_over_configuration
https://maven.apache.org
https://maven.apache.org/guides/introduction/introduction-to-the-lifecycle.html
https://maven.apache.org/guides/introduction/introduction-to-plugins.html
https://maven.apache.org/guides/introduction/introduction-to-archetypes.html
https://www.sonatype.com/ebooks
https://pragprog.com/book/auto/pragmatic-project-automation
https://www.jetbrains.com/lp/devecosystem-2021/java/