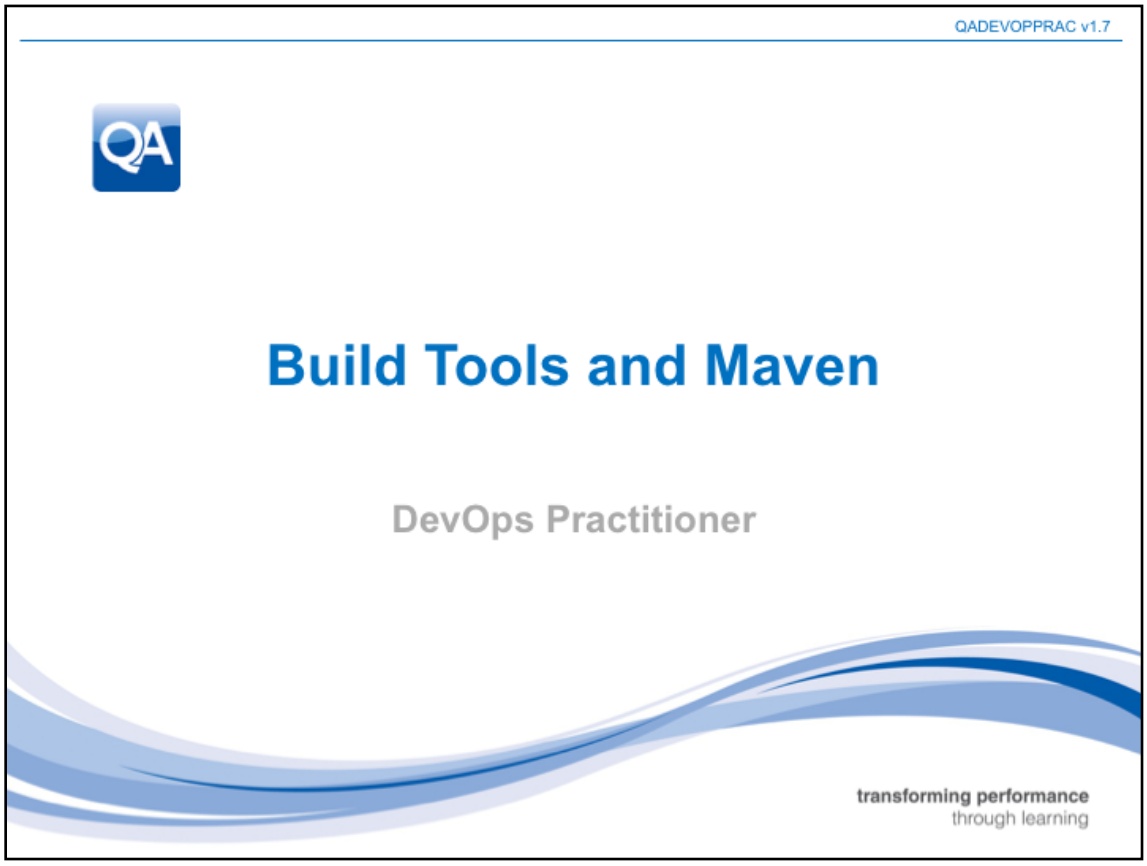
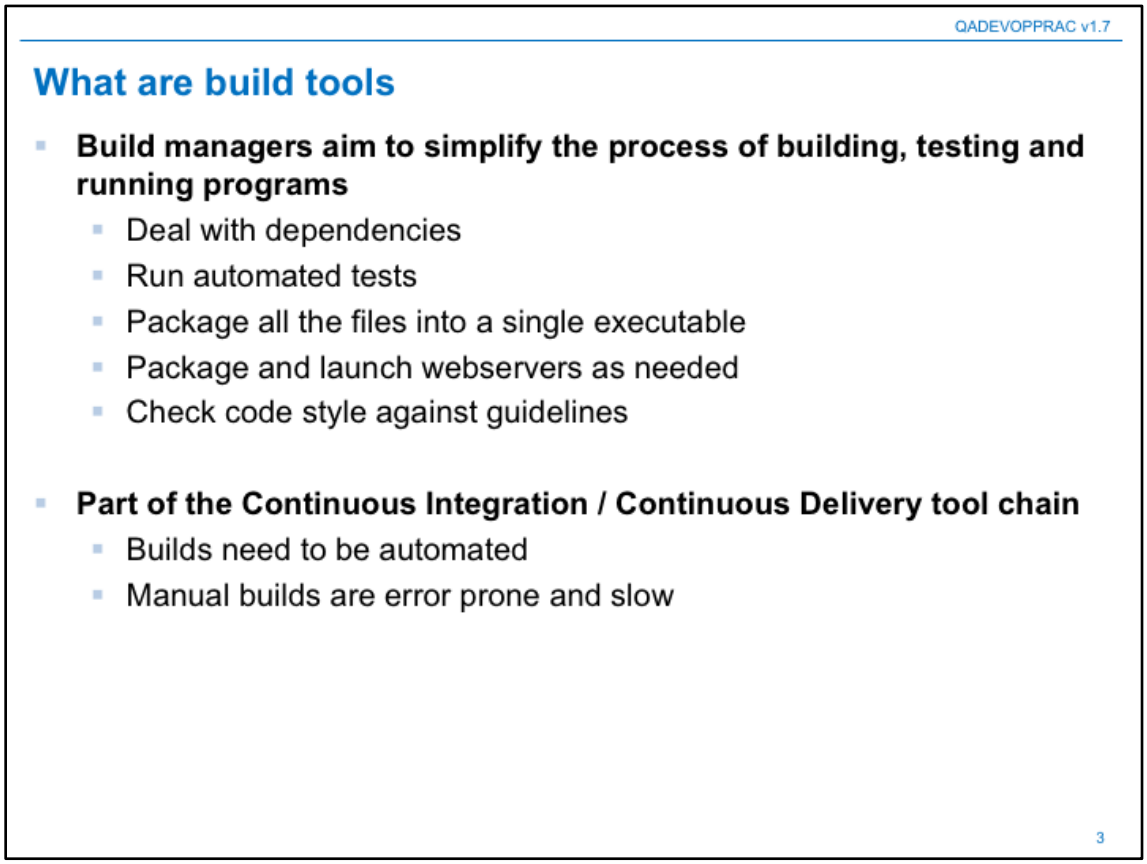

QADEVOPPRAC v1.7
Build Tools and Maven
DevOps Practitioner
transforming performance
through learning

Outline

- **What are build tools**
  - Why are they useful
  - Different build tools available
- **Introducing Maven**
  - What is Maven
  - Why do we like it
  - Maven Principles
- **Installing Maven**
  - Pulling down an archetype
  - Running a project



Wikipedia's List: http://en.wikipedia.org/wiki/List_of_build_automation_software

Prevence us from a lot of bad things happening: software version changed, Plugin is missing, library can change etc. So process is very unstable. Tools will make the process stable.

Deal with dependencies – finds libraries and makes sure, that they have same version.

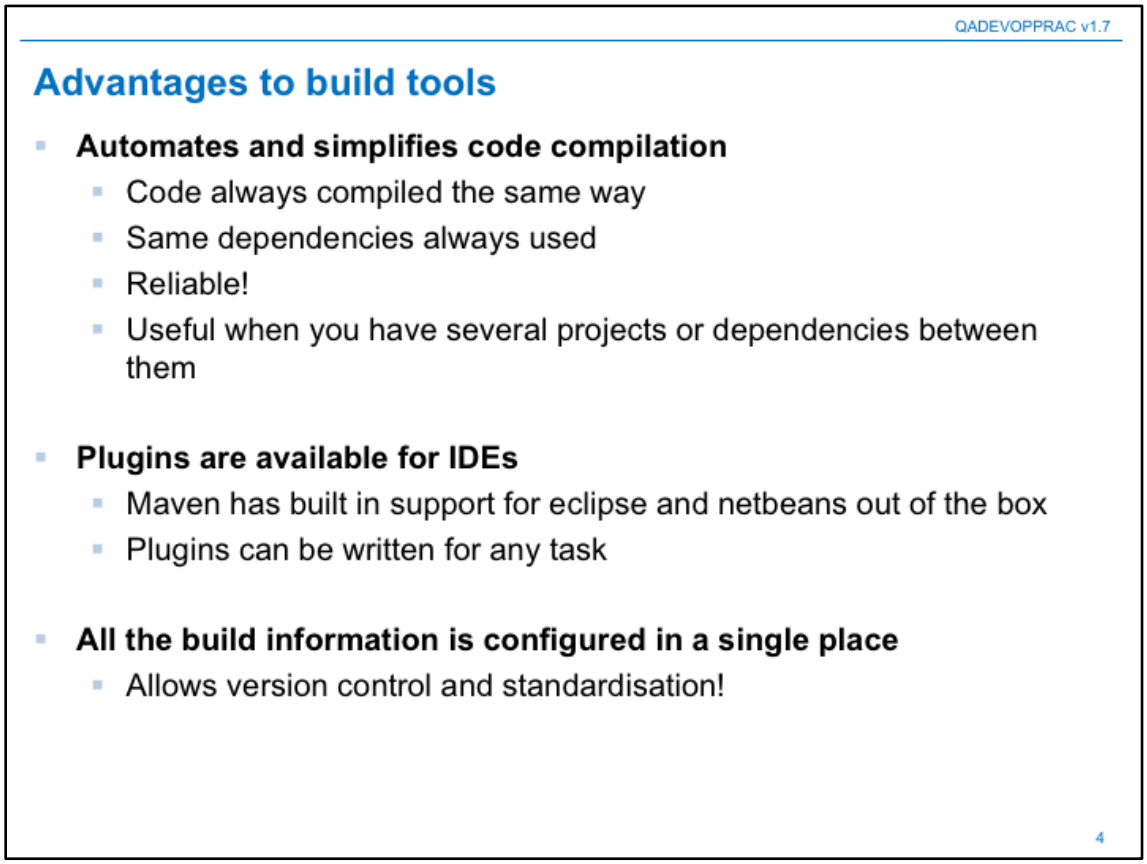
Run automated tests – people don't like boring things. Test can be done automatically.

Package all the files into a single executable – automatically can build jar or war

Package and launch webservers as needed – possible to build and launch web server from a command line

Check code style against guidelines – can check is submitted code is build following guidelines

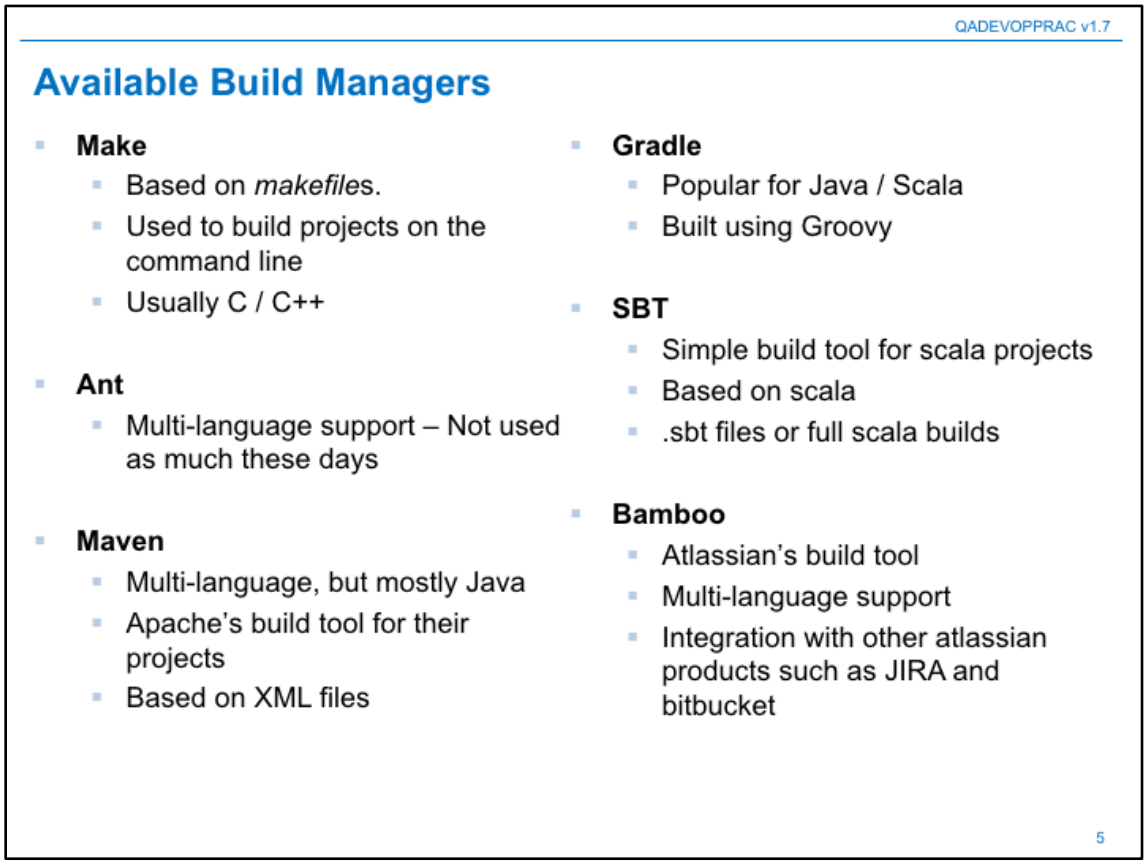
Part of the Continuous Integration / Continuous delivery



Reliable – no “it worked on my machine”. No it works on any machine: windows, mac, linux etc.

Repeatable code compiled the same way from same dependencies.

Useful when you have several projects or dependencies



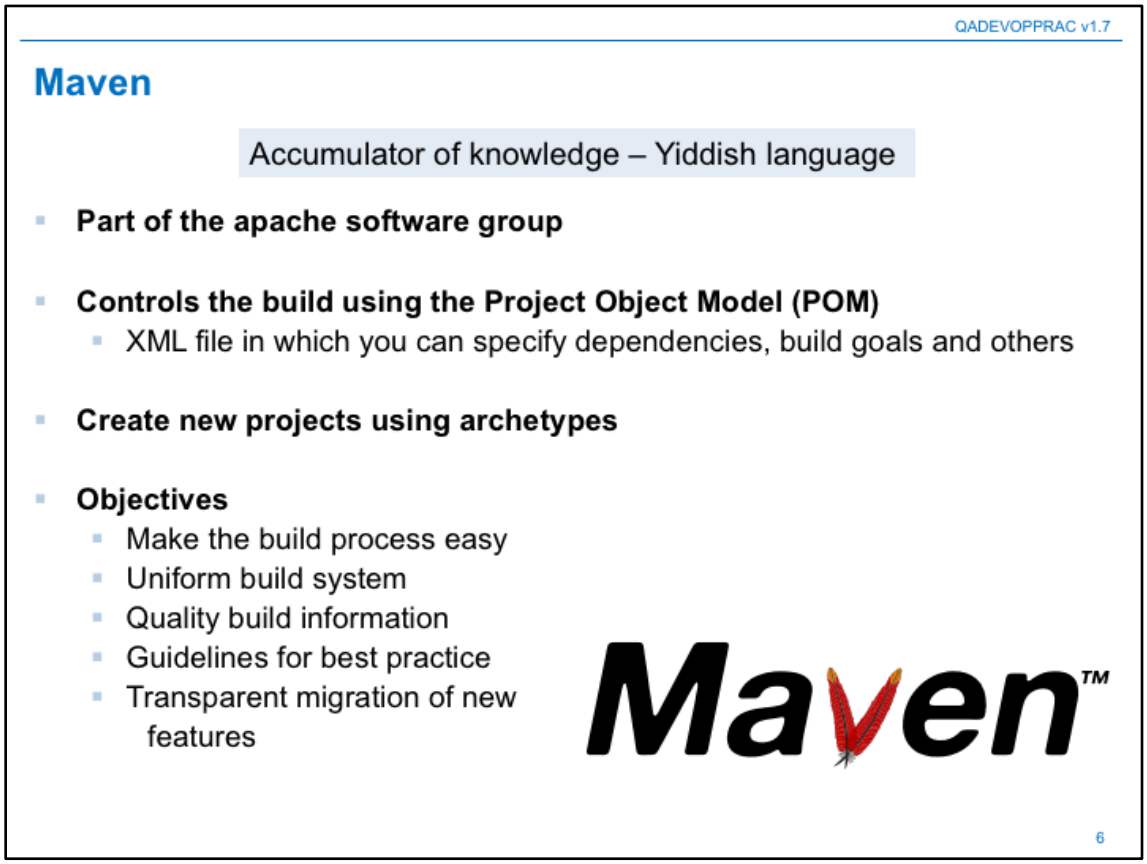
Wikipedia's List: http://en.wikipedia.org/wiki/List_of_build_automation_software

There are many build tools:

Ant – for multi-language projects

Maven and Gradle – popular for Java projects : do similar jobs but describe it different ways (maven – xml based, Gradle – groovie based)

SBT for Scala



Objectives stated on: https://maven.apache.org/what-is-maven.html (18/5/2015)

Started as part of apache software group

Apache builded very large and complicated projects and it becomes out of hands.

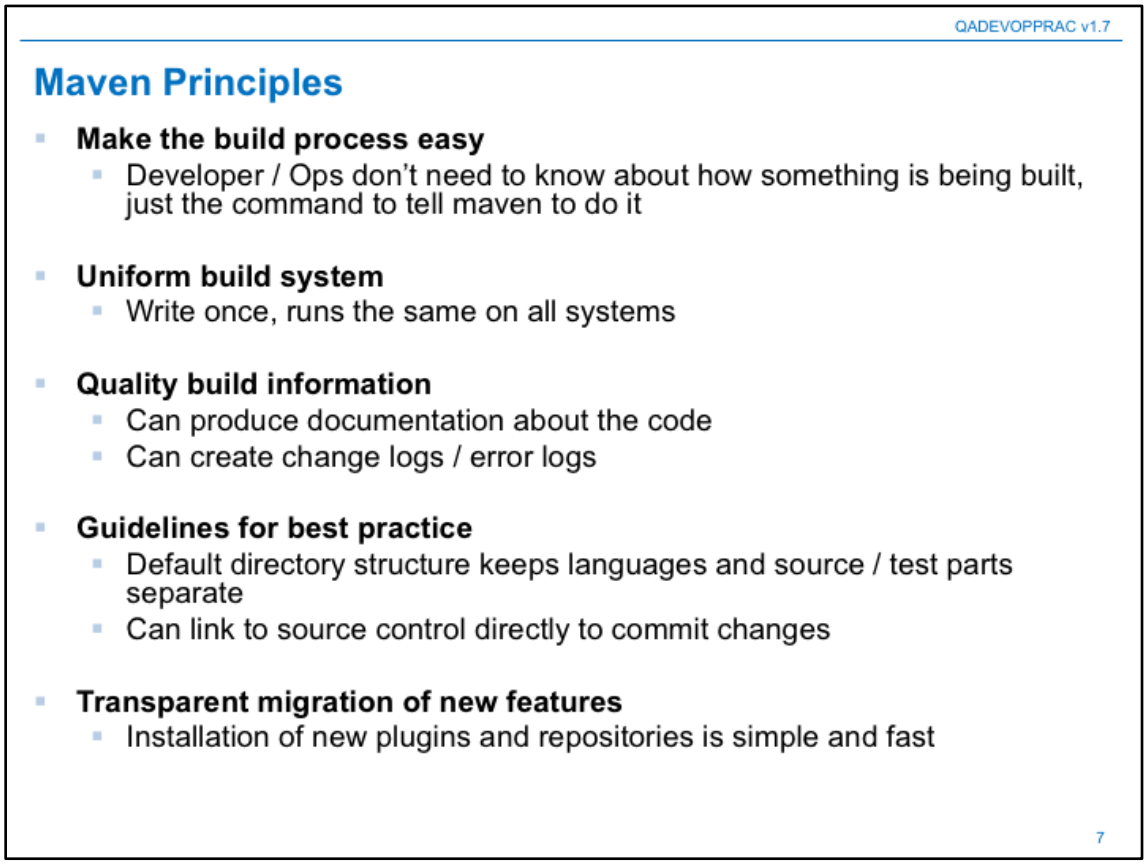
Apache foundation find the way how to describe and to build it. And they started to do this with Java projects.

Now it can be anything

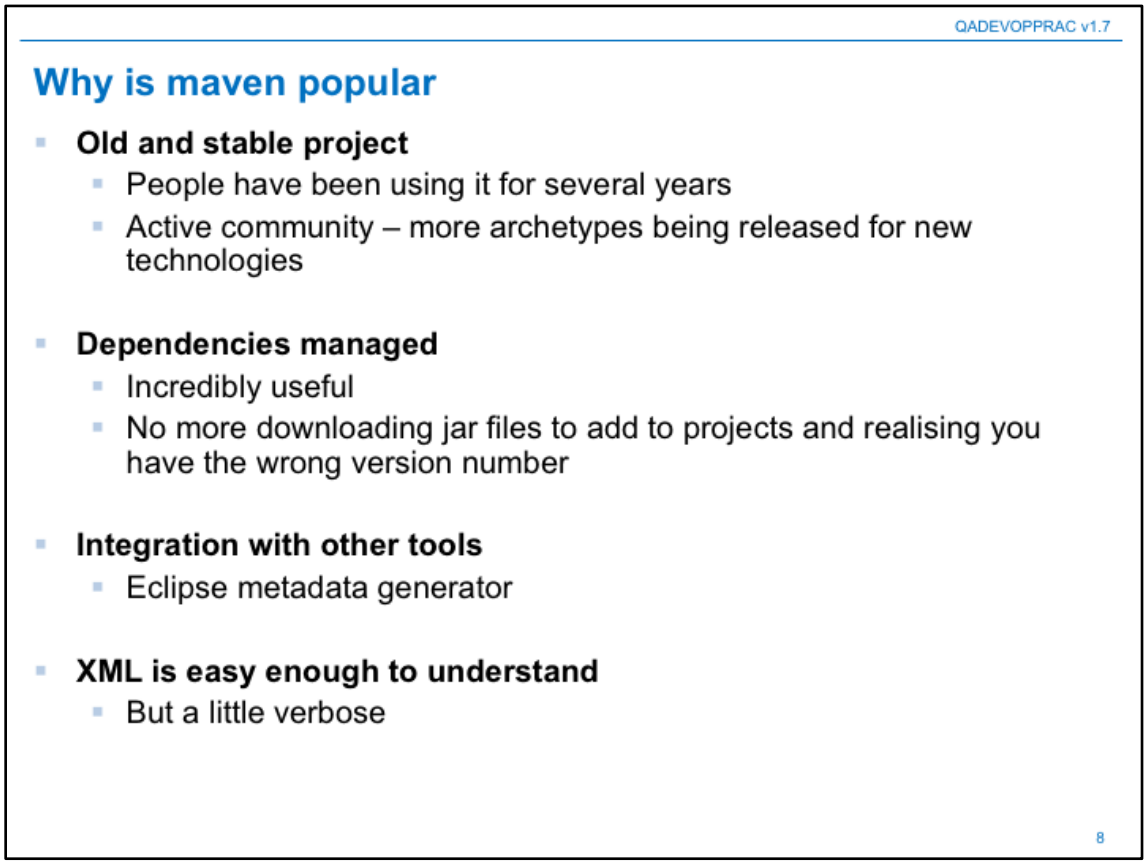
Everything is controlled from the single point – Project Object Model (POM file)

Here you can put in goals, dependencies and source controls management you want.

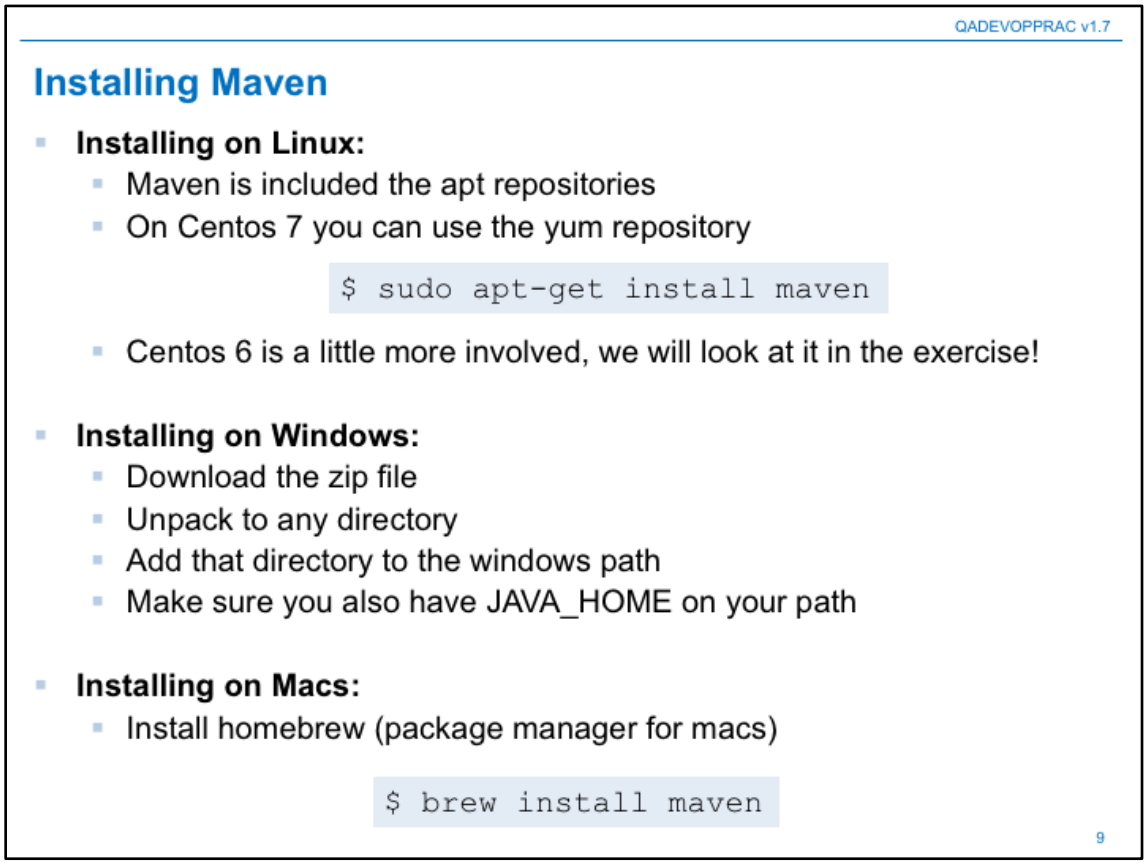
When you start a new project you can ask Maven to get a skeleton for this. Its all build into one.



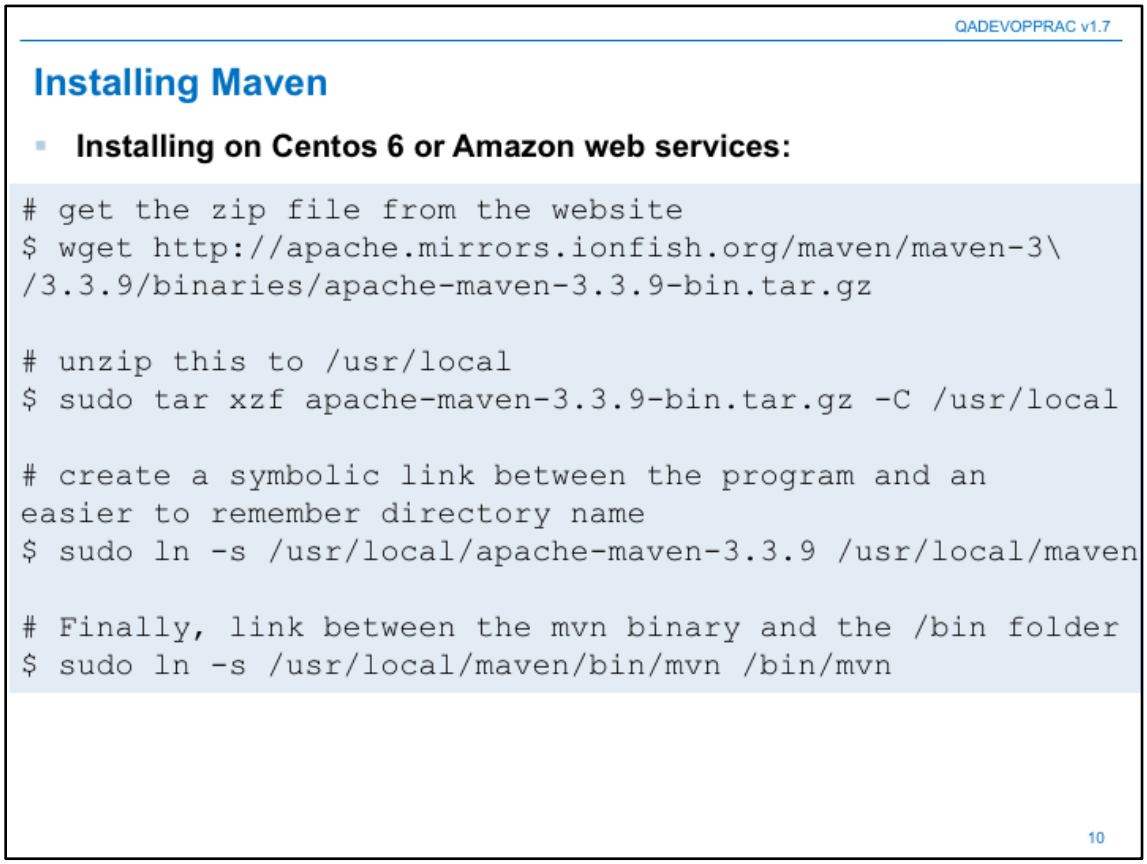
- Make the build process easy – in the command line its MVN GOAL (it does not matter what, maven will go and reach a goal.
- Uniform build system – it does not matter where you build your system – it will run the same way on another as well
- Quality build information – designed to give outputs after building some your project - forms, dashboard etc. – says what was happen with your software and is something needs to be changed in terms of testing
- Guidelines for best practice – there are a lot of guidelines of Maven just by lucking basic setup for the project. Source control goes to the src folder and test – to test folder. You can change these folders, but traditionally is build for every one (he can look and knows how it works).
- Transparent migration of new features – you can install new plugins and repositories just by adding something to your pom file.



A lot of people really don't like maven, but it is still going strong despite this. Gradle works in a similar way to maven but it is not XML based and considered to be more flexible.



On Centos 6 and Amazon:



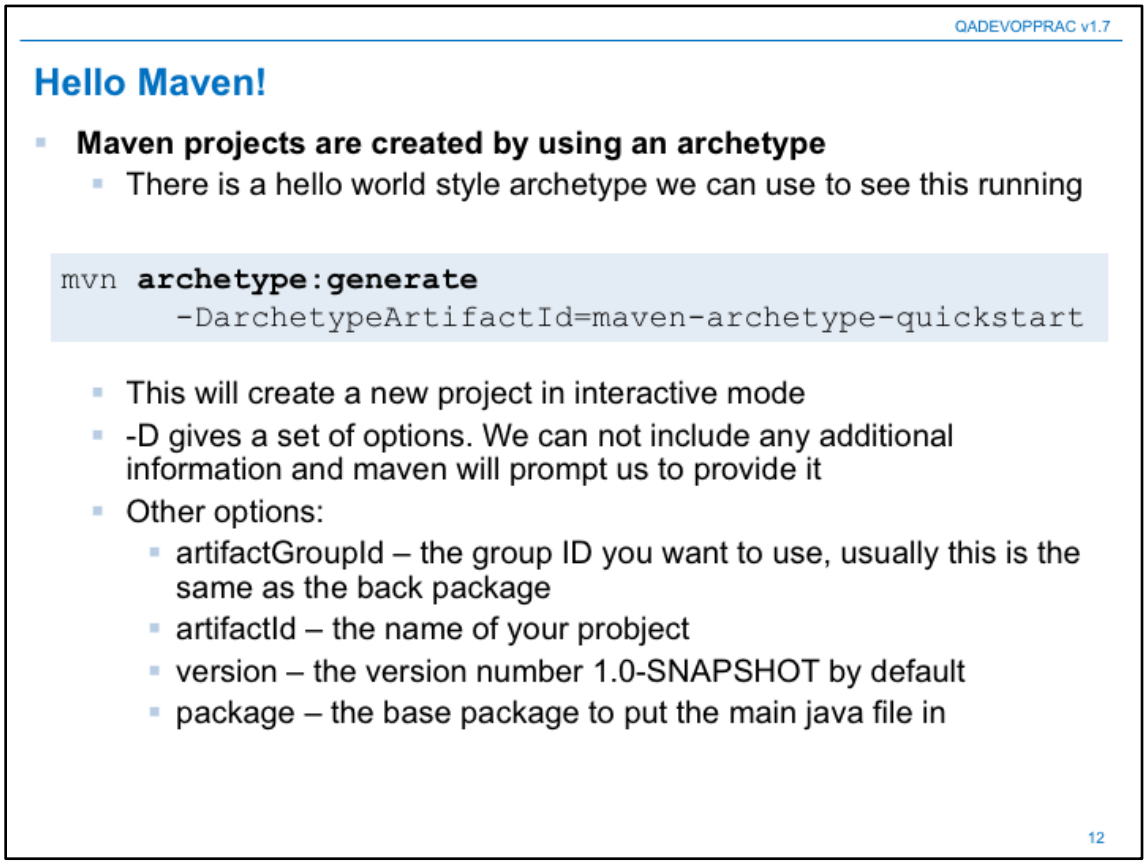
On Centos 6 and Amazon:

Hello Maven!

- **Now we can see if maven is running on our system**

```
$ mvn --version
Apache Maven 3.2.5
Maven home: /usr/local/Cellar/maven/3.2.5/libexec
Java version: 1.8.0_25, vendor: Oracle Corporation
Java home: /Library/Java/JavaVirtualMachines/
                        jdk1.8.0_25.jdk/Contents/Home/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "mac os x", version: "10.10.3", arch: "x86_64",
                                          family: "mac"
```

The output will change based on which operating system you have and which version number of the software you have!



All options:

https://maven.apache.org/archetype/maven-archetype-plugin/generate-mojo.html

mvn artifact:generate

mvn compile

The Directory Structure

- **Default for maven projects**

```
my-app
   - pom.xml
   - src
      - main
         -java
            - com
               - qa
                  -myapp
                     - App.java
      - test
         - java
            - com
               - qa
                  - myapp
                     - AppTest.java
```

`<project xmlns="...">` - namespace of your project

`<modelVersion>4.0.0` – this will always be same number!!!!!!!

`<url>http://maven.apache.org` – you can give it and it will be shown in metadata

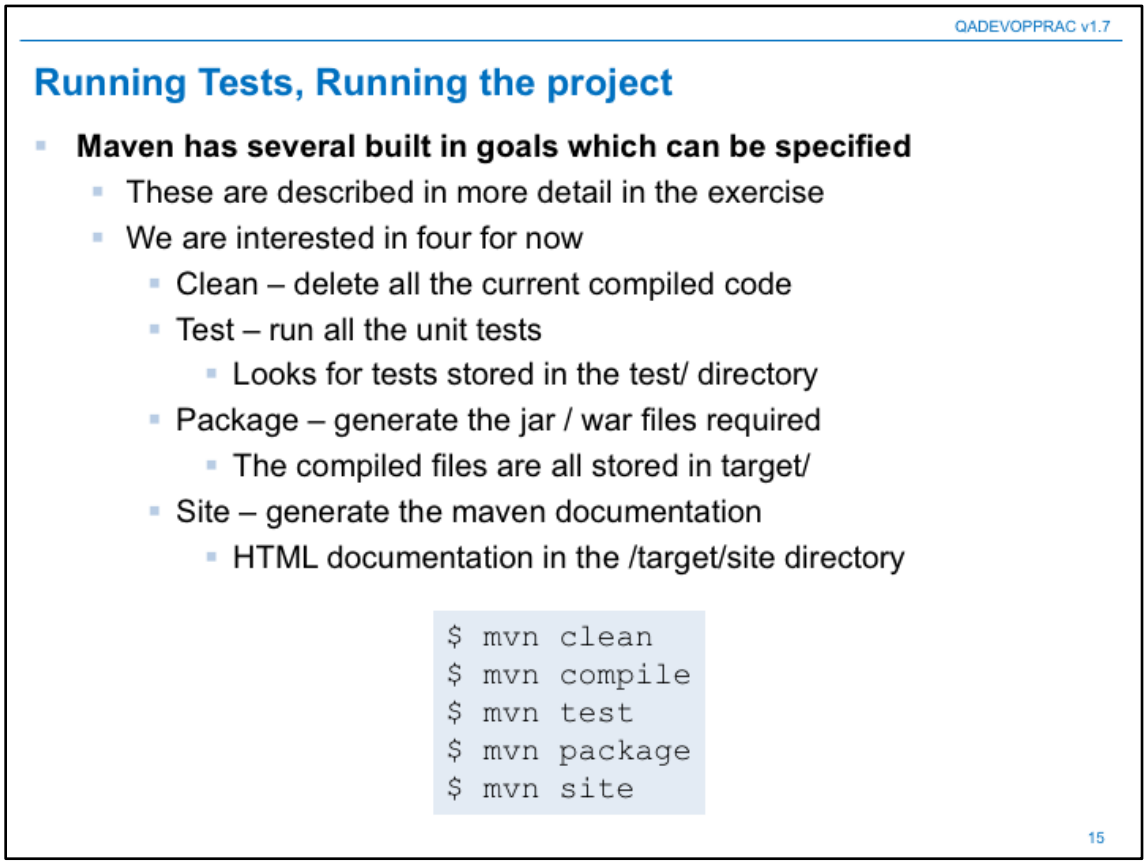
`<dependency>`

includes junit only for tests – so it will be used only in the place where its required (not in all project)

The groupID, artifactID, version and name were all generated from the values we gave maven

The model version is based on the archetype

A list of dependencies tells maven which additional jar files are required. The scope of <test> means that it is only included when the project is being tested, rather than when it is packaged up for final release.



To run my-app project

cd target/classes

java com.qa.App

Or

java -cp target/classes com.qa.App

The Eclipse Plugin

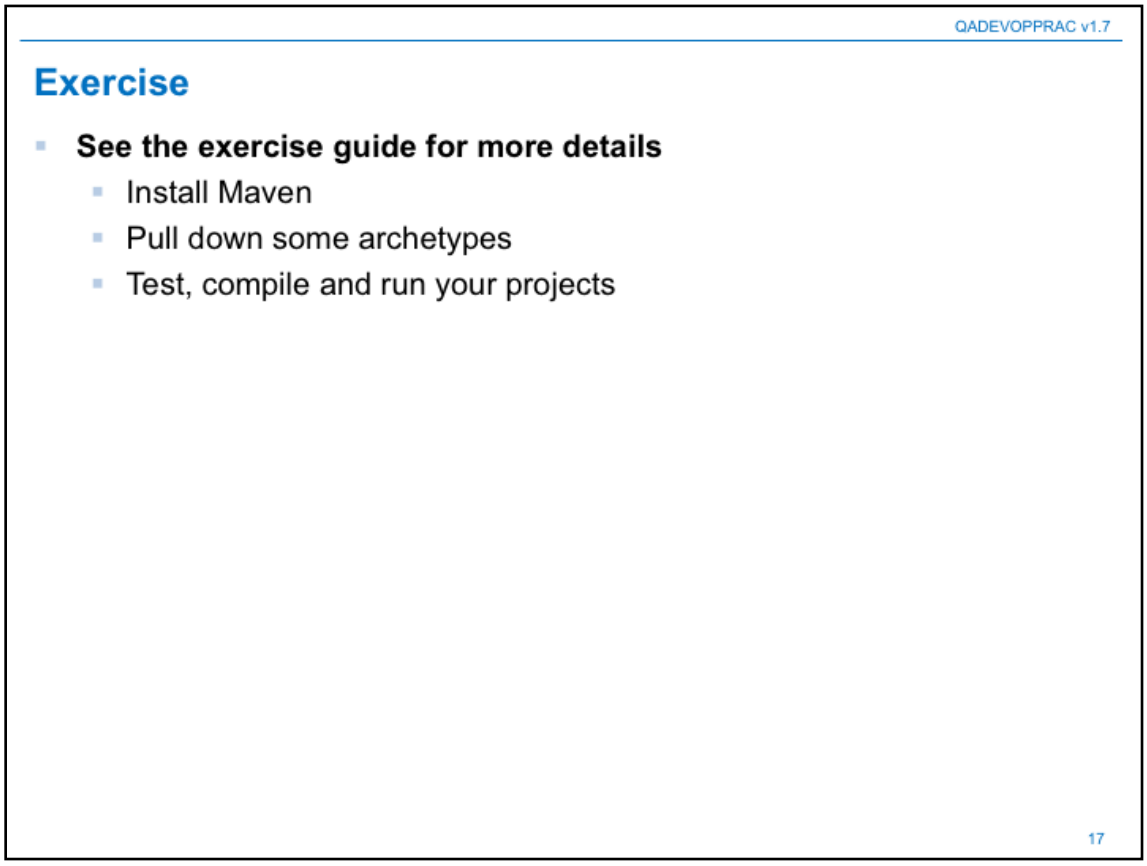
- To generate all the eclipse metadata run the following command

```
$ mvn eclipse:eclipse
```

- We can also clean out the metadata if required

```
$ mvn eclipse:clean
```

- Projects can be imported directly from the pom by using eclipses' import


QADEVOPPRAC v1.7
Exercise
See the exercise guide for more details
Install Maven
Pull down some archetypes
Test, compile and run your projects
17

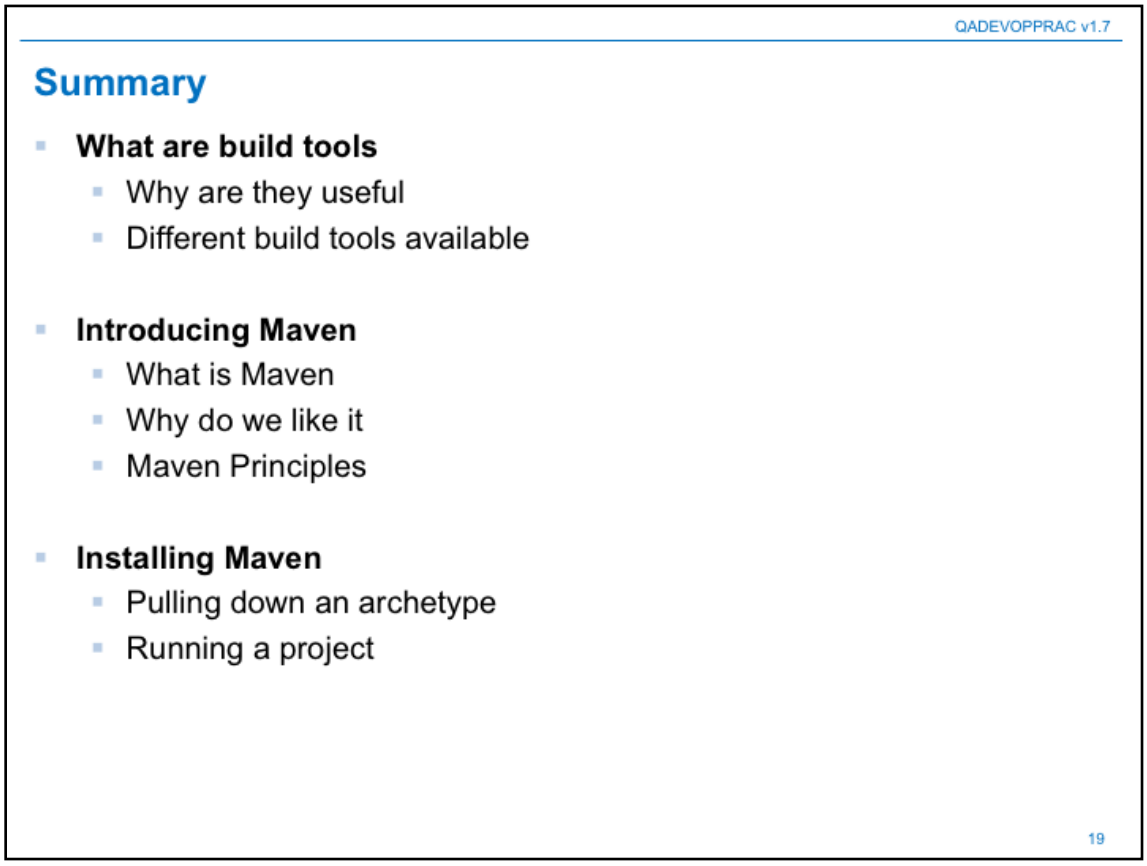
To read more about Maven

Books:

- Apache Maven Cookbook (2015. link: https://www.packtpub.com/application-development/apache-maven-cookbook)
- Maven Crash Course (2017. link: https://www.safaribooksonline.com/library/view/maven-crash-course/9781787124912/)
- Learning Apache Maven (2015. link: https://www.safaribooksonline.com/library/view/learning-apache-maven/9781771373661/)
- Apache Maven Project books (2017. link: https://maven.apache.org/articles.html)

Tutorials:

- https://maven.apache.org/guides/getting-started/maven-in-five-minutes.html
- https://www.udemy.com/mavencrashcourse/
  https://www.tutorialspoint.com/maven/index.htm


QADEVOPPRAC v1.7
Summary
What are build tools
Why are they useful
Different build tools available
Introducing Maven
What is Maven
Why do we like it
Maven Principles
Installing Maven
Pulling down an archetype
Running a project
19