Software Project Management - Laboratory

Lecture n° 21
A.Y. 2020-2021

Prof. Fabrizio Fornari
fabrizio.fornari@unicam.it

Fill the evaluation questionnaire

https://www.unicam.it/studente/questionari-sulla-didattica

Course Overview

Course Objective

The course introduces the students to the basic knowledge of complex software system production following the **DevOps methodology**.

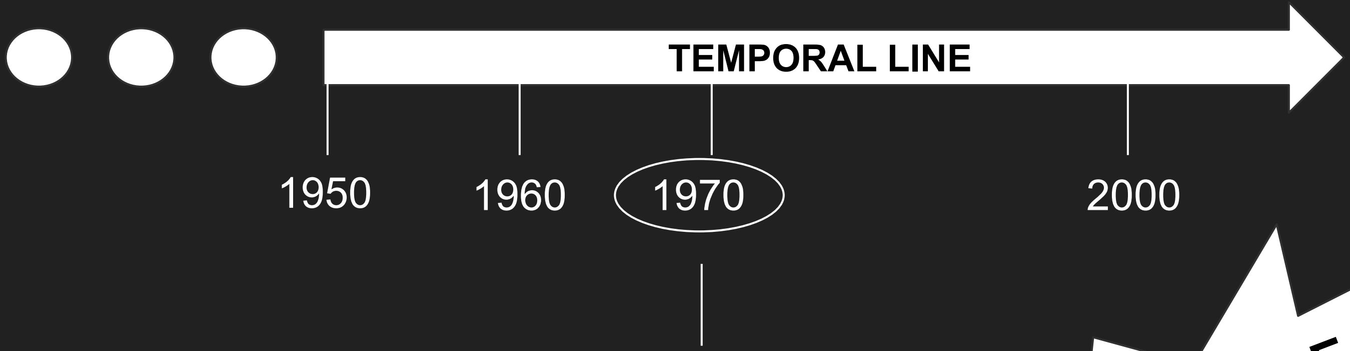
Prerequisite knowledge

- Basic Programming Experience
- Basic Software Engineering Methods and Techniques

Learning Outcome

The student will be able to manage the organization and the development of a software applying DevOps methodology.

Lecture 1



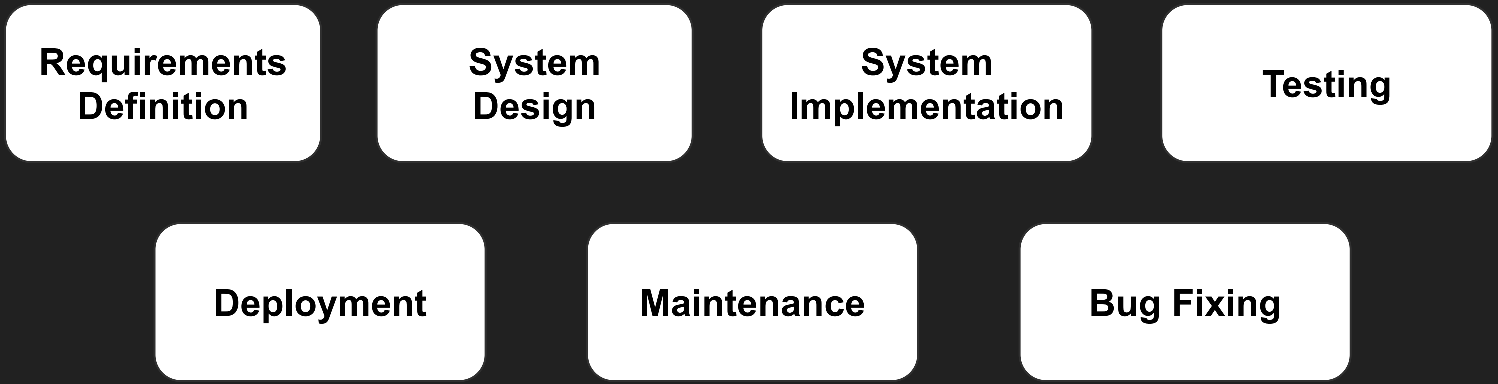
- Late ‘60s
- Projects running out of budget
- Projects running late
- Low Quality Software
- Software not compliant with the requirements
- Unmanageable projects, code too difficult to maintain

SOFTWARE CRISIS

An answer to the Software Crisis

- Recognising that developing software is a complex process similar to those that generates engineering products (Software Development Process)

- The birth of Software Engineering



Requirements Definition

System Design

System Implementation

Testing

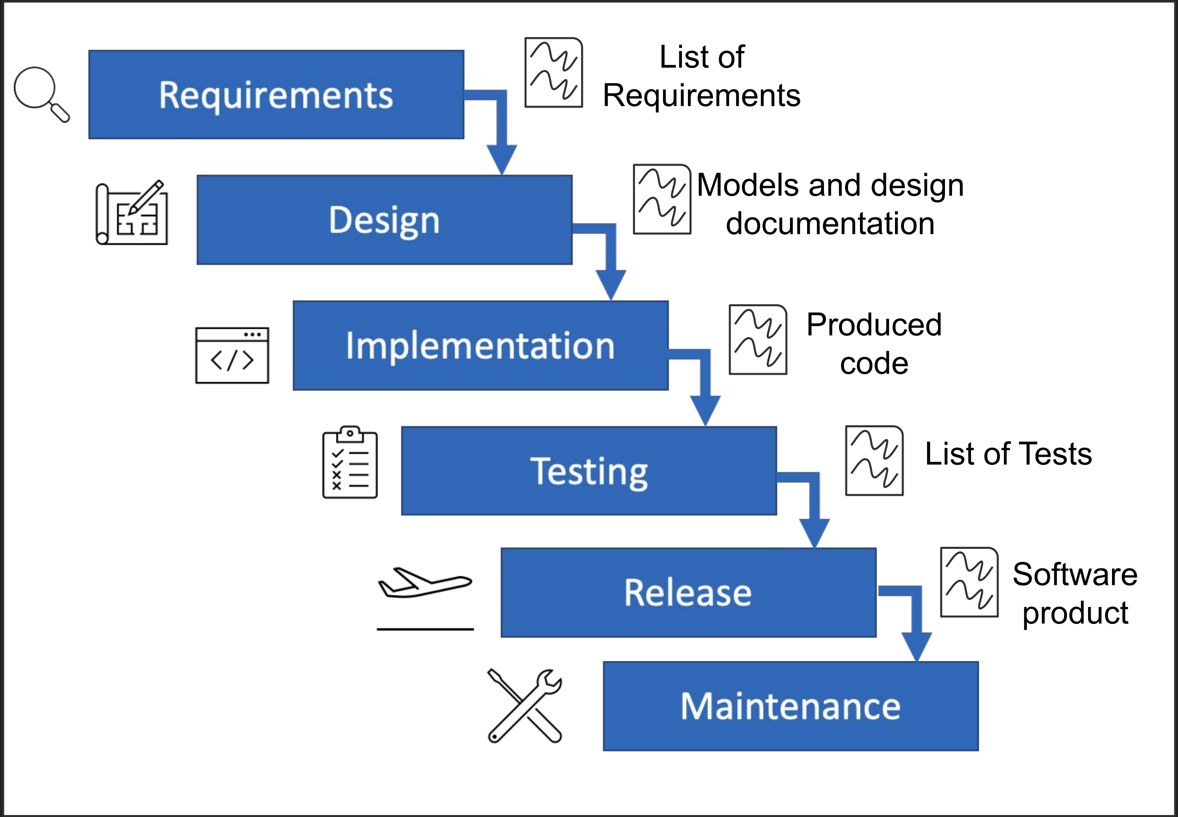
Deployment

Maintenance

Bug Fixing

Waterfall Model

- Guided by the production of documents
- Progress measurable based on the amount of documentation produced
- Documents to support personnel changes

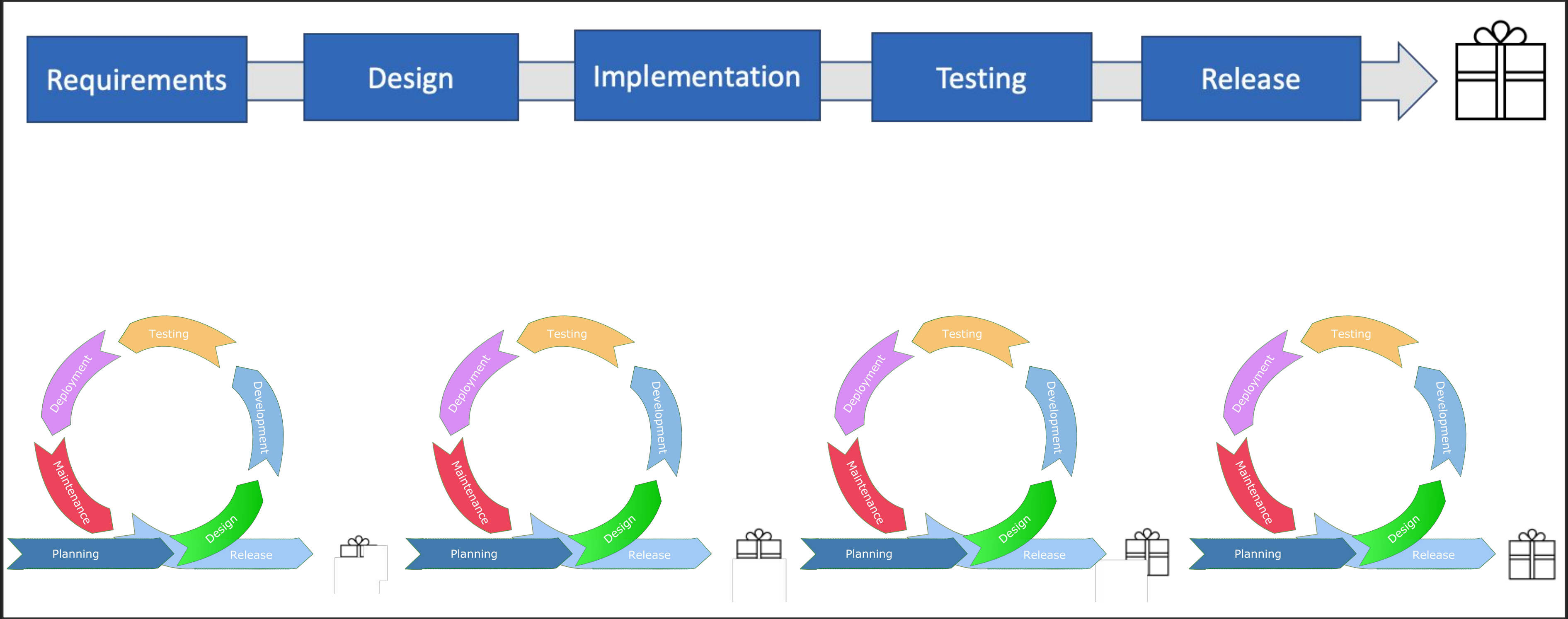


The Agile Manifesto

Individuals Interactions	>	Processes Tools
Working Software	>	Comprehensive Documentation
Customer Collaboration	>	Contract Negotiation
Responding to Change	>	Following a Plan

Manifesto: https://agilemanifesto.org/

Waterfall vs Agile



Requirements → Design → Implementation → Testing → Release

Testing
Development
Design
Release
Planning
Maintenance
Deployment

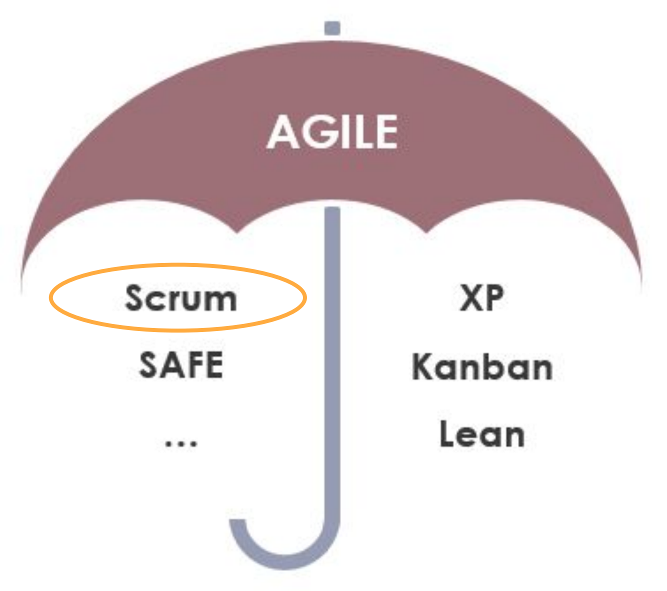
Testing
Development
Design
Release
Planning
Maintenance
Deployment

Testing
Development
Design
Release
Planning
Maintenance
Deployment

Testing
Development
Design
Release
Planning
Maintenance
Deployment

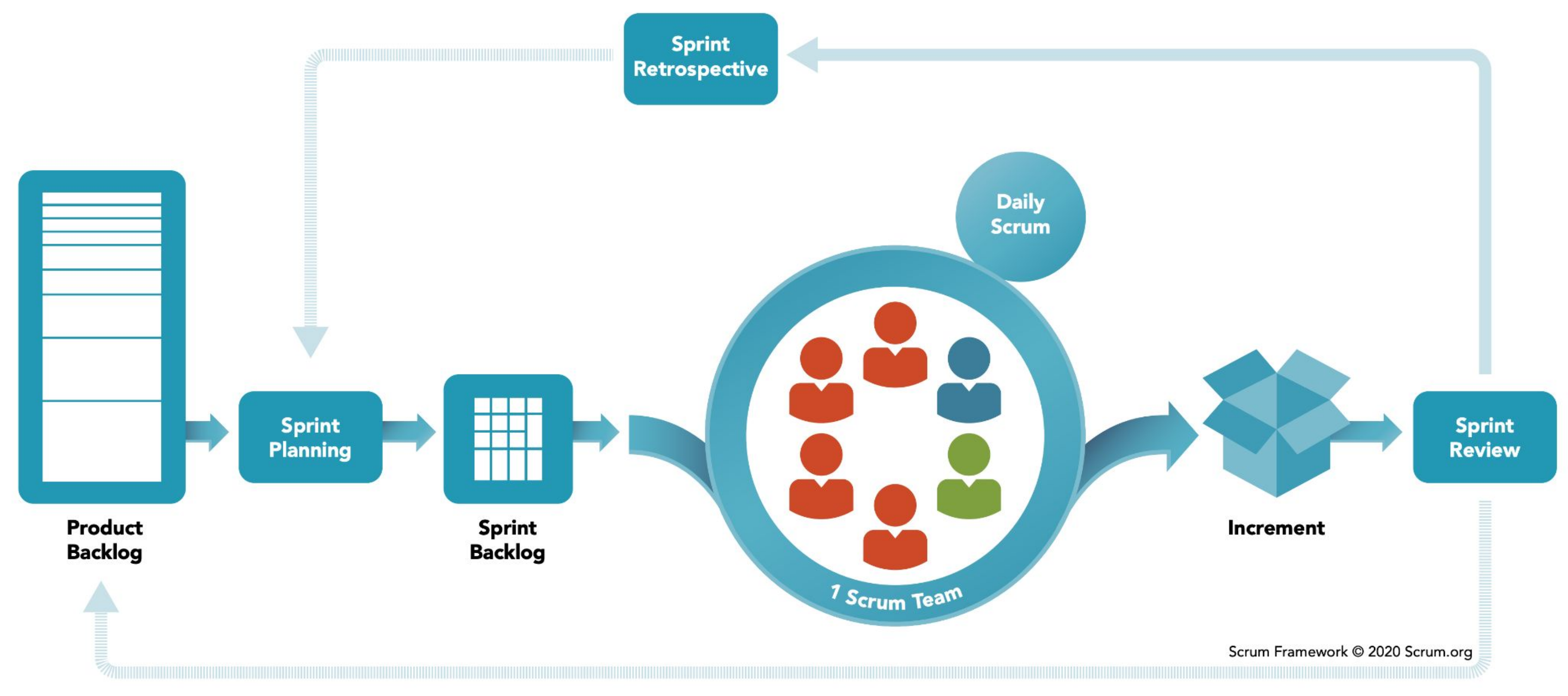
SCRUM

Scrum is an Agile framework for project management that emphasizes teamwork, accountability and iterative progress toward a well-defined goal.



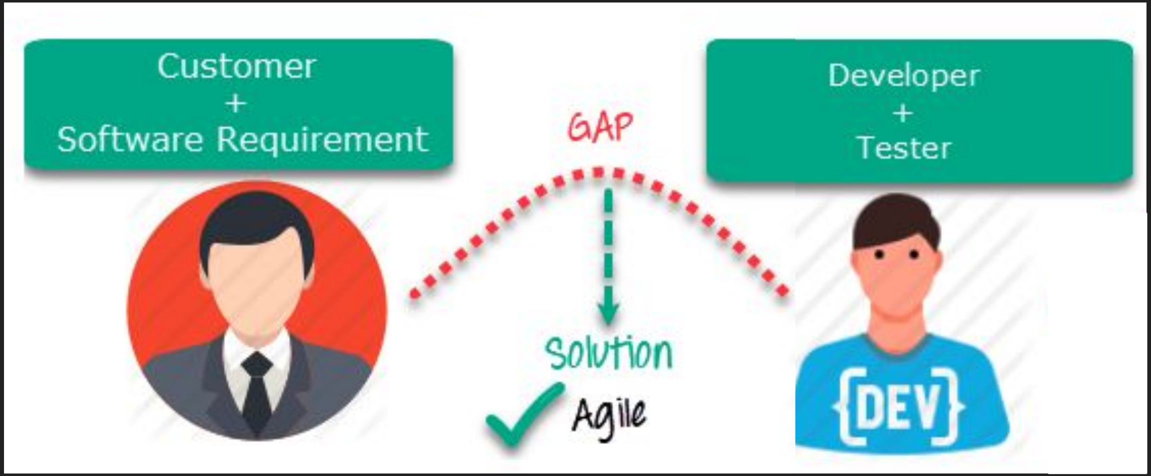
Schwaber, K. (1997). Scrum development process. In *Business object design and implementation* (pp. 117-134). Springer, London.

SCRUM - Framework

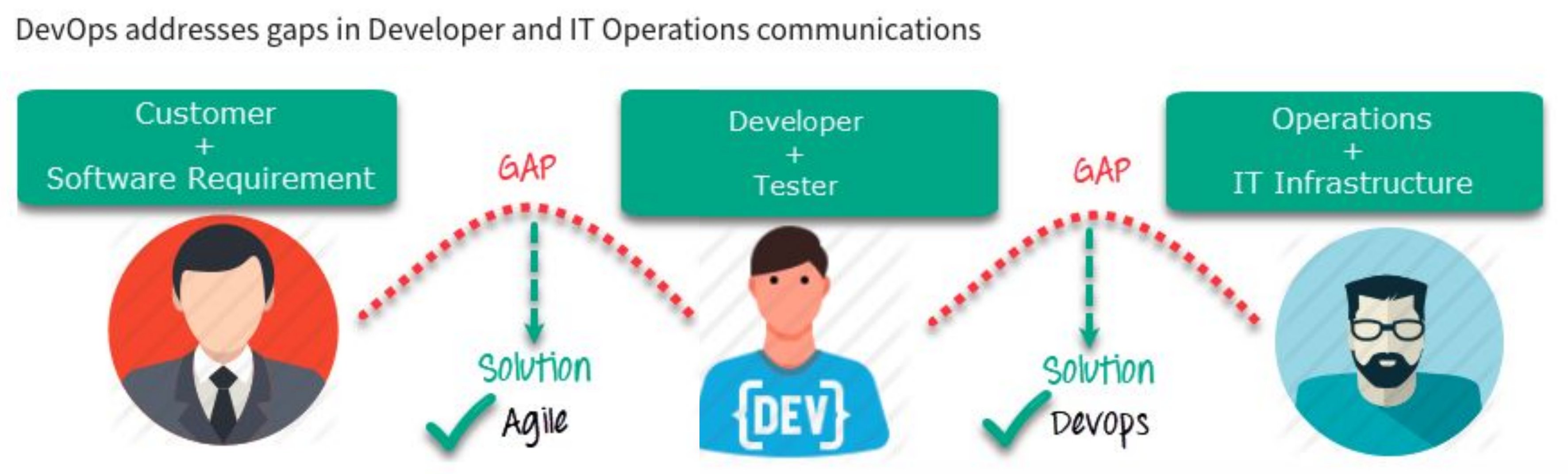


Scrum Framework © 2020 Scrum.org

Focus of Agile paradigm

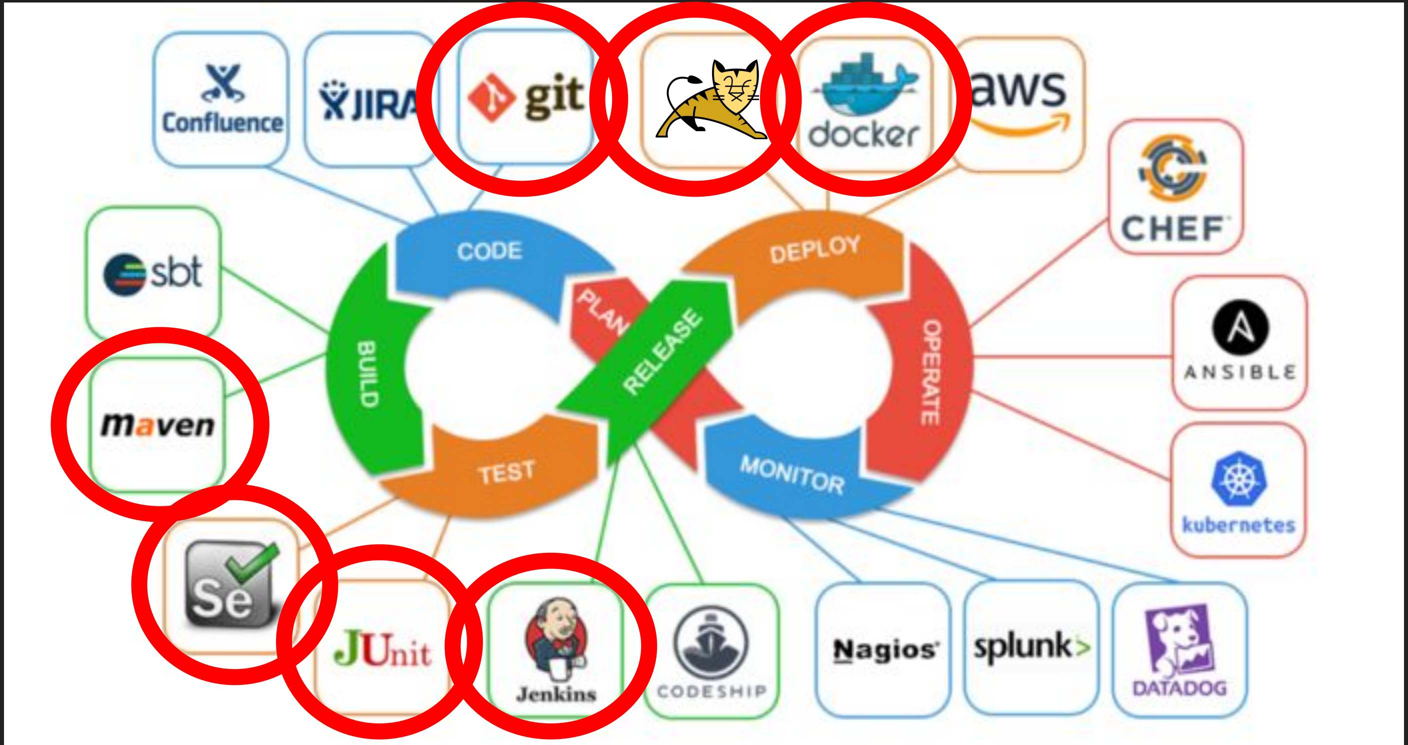


DevOps

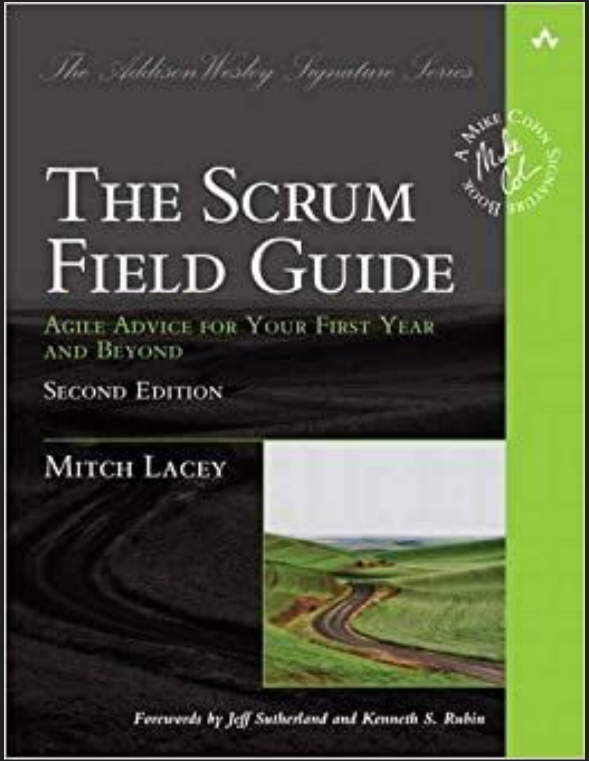


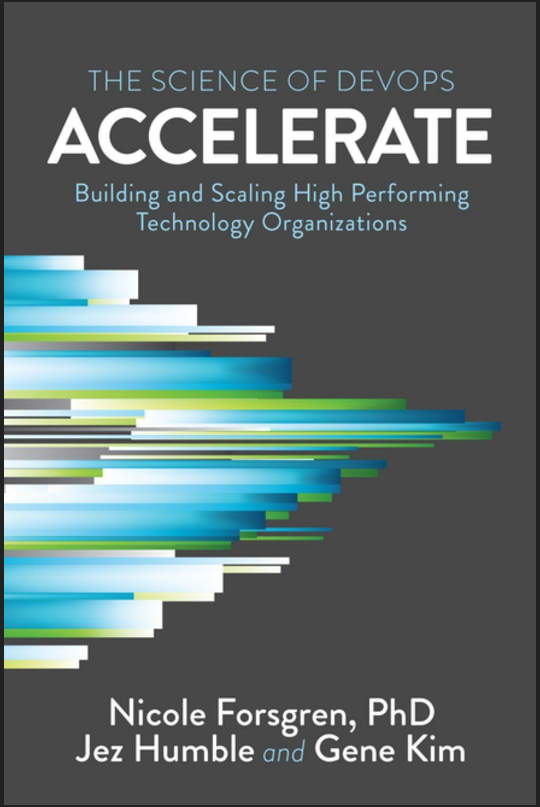
DevOps addresses gaps in Developer and IT Operations communications

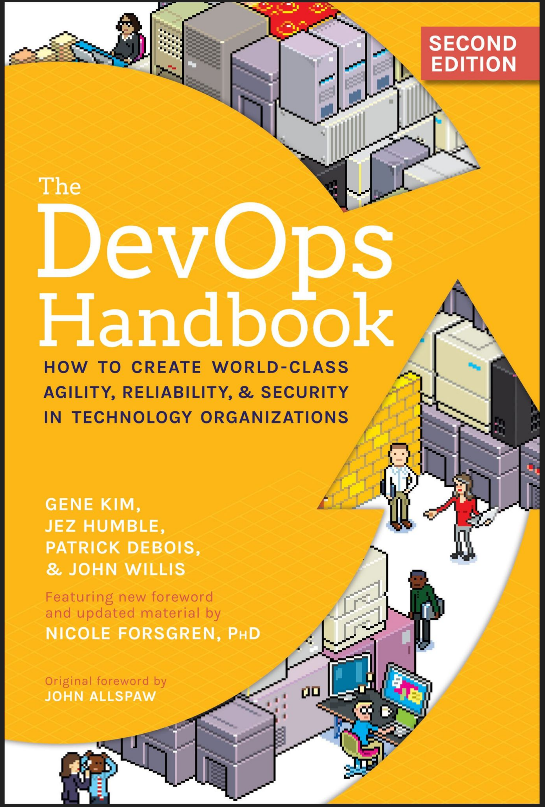
DevOps


Confluence
JIRA
git
docker
aws
CHEF
sbt
ANSIBLE
maven
kubernetes
Se
JUnit
Jenkins
CODESHIP
Nagios
splunk>
DATADOG
CODE
PLAN
RELEASE
DEPLOY
OPERATE
BUILD
TEST
MONITOR

Some References


The Addison Wesley Signature Series
A Mike Cohn Signature Book
The Scrum Field Guide
Agile Advice for Your First Year and Beyond
Second Edition
Mitch Lacey
Forewords by Jeff Sutherland and Kenneth S. Rubin



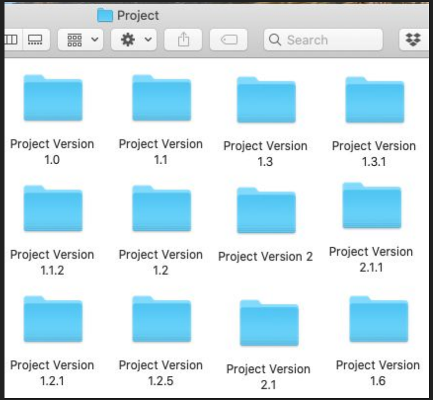


Lecture 2

Version control is a system that records changes to a file or set of files over time so that you can recall specific versions

How to do it?

Manually



Local Version Control Systems

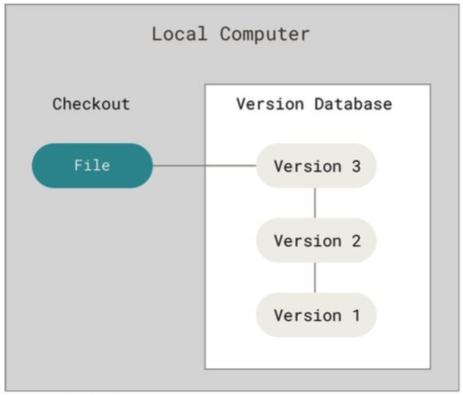


Figure 1. Local version control

Distributed Version Control Systems

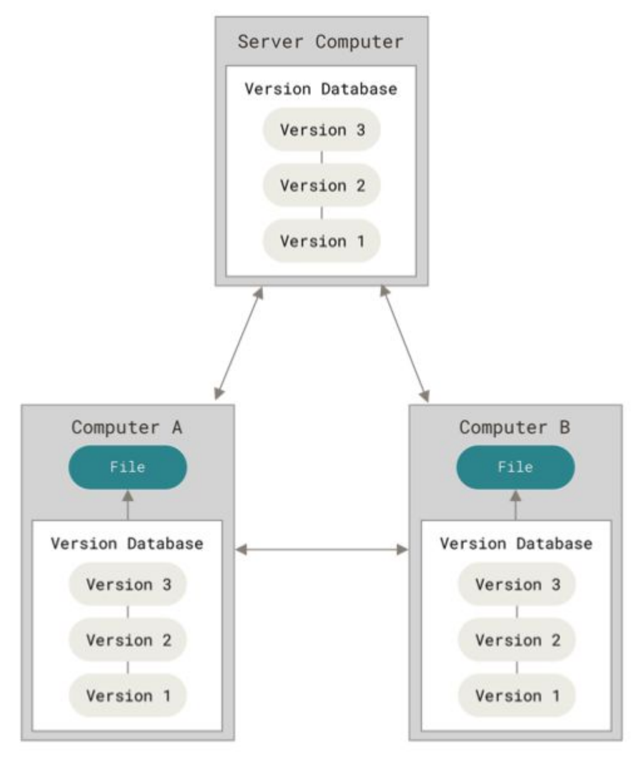


Figure 3. Distributed version control

Centralized Version Control Systems

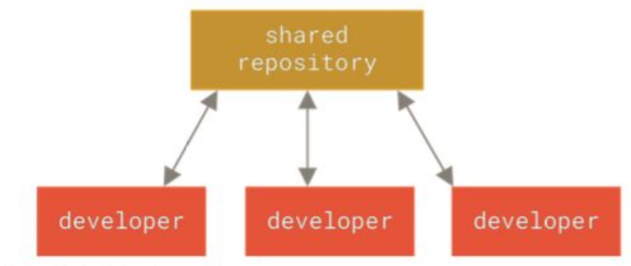


Figure 2. Centralized version control

What is git?

- A distributed version control system - DVCS. It means that there is no main server and all of the full history of the project is available once you cloned the project.
- Open source project originally developed in 2005 by Linus Torvalds
- A command line utility
- You can imagine git as something that sits on top of your file system and manipulates files.

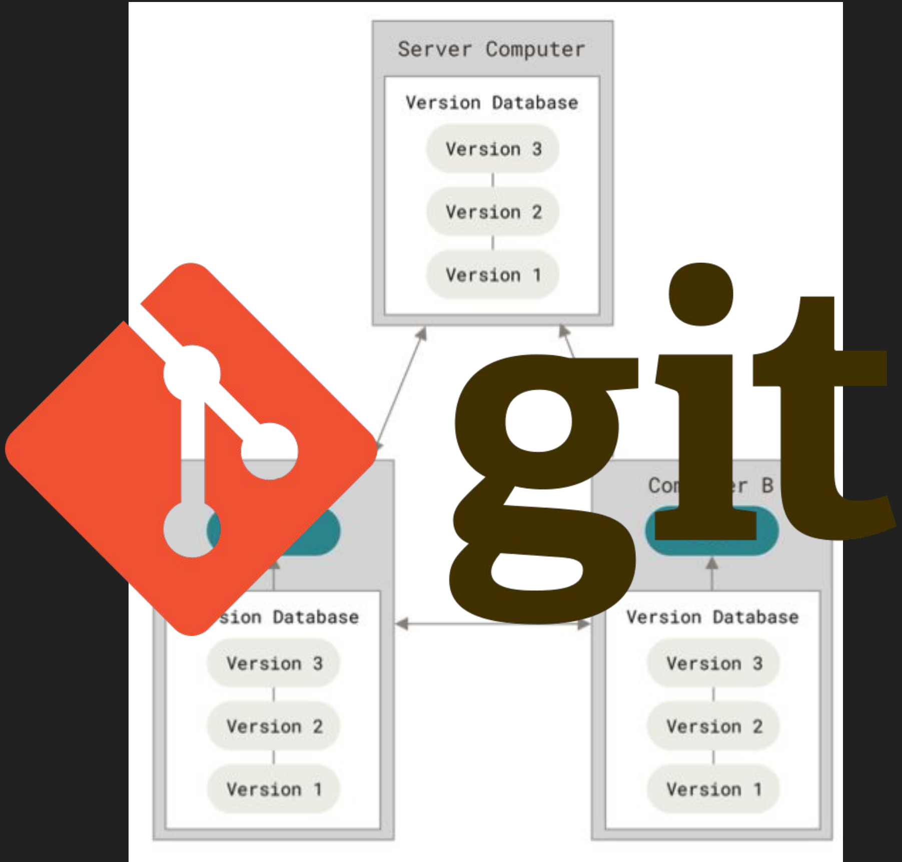


Figure 3. Distributed version control

https://git-scm.com/

Git - Three Sections

Three main sections of a Git project: the working tree, the staging area, and the Git directory.

Git Workflow

1. Modify file in working directory
2. Stage changes you want to commit
3. Commit, takes the file as they are in the saging area and stores that snapshot permanently to your Git directory

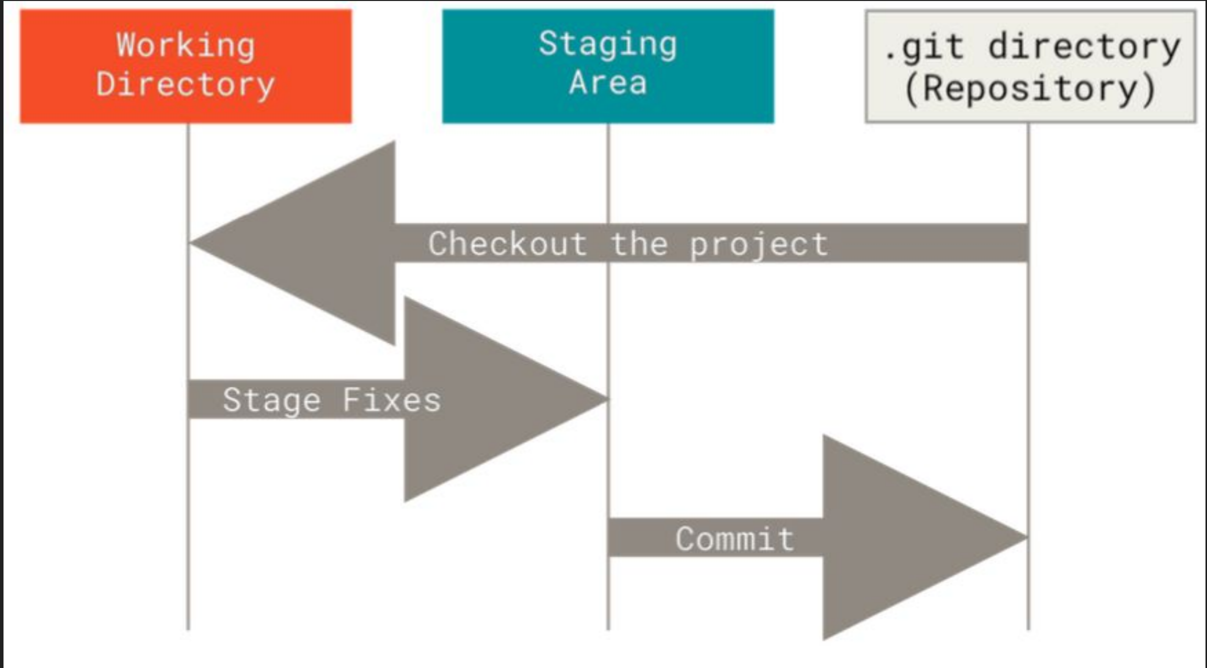
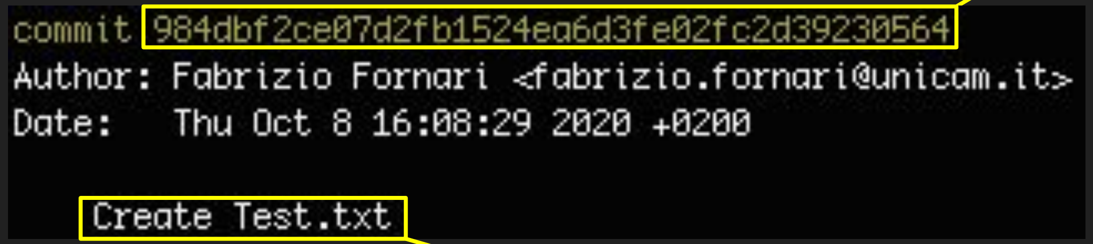


Figure 6. Working tree, staging area, and Git directory

Git - Commit

A commit object mainly contains three things:

- A hash, a 40-character string that uniquely identifies the commit object
- Commit message describing the changes
- A set of changes the commit introduces



```
commit 984dbf2ce07d2fb1524ea6d3fe02fc2d39230564
Author: Fabrizio Fornari <fabrizio.fornari@unicam.it>
Date:   Thu Oct 8 16:08:29 2020 +0200

    Create Test.txt
```

Commit id (hash)

Commit message

Let’s start!

1. Check if you have a version of git installed on your machine $git --version
2. If not, install it https://git-scm.com/book/en/v2/Getting-Started-Installing-Git
3. Set your user name and email address; every Git commit will use this information.

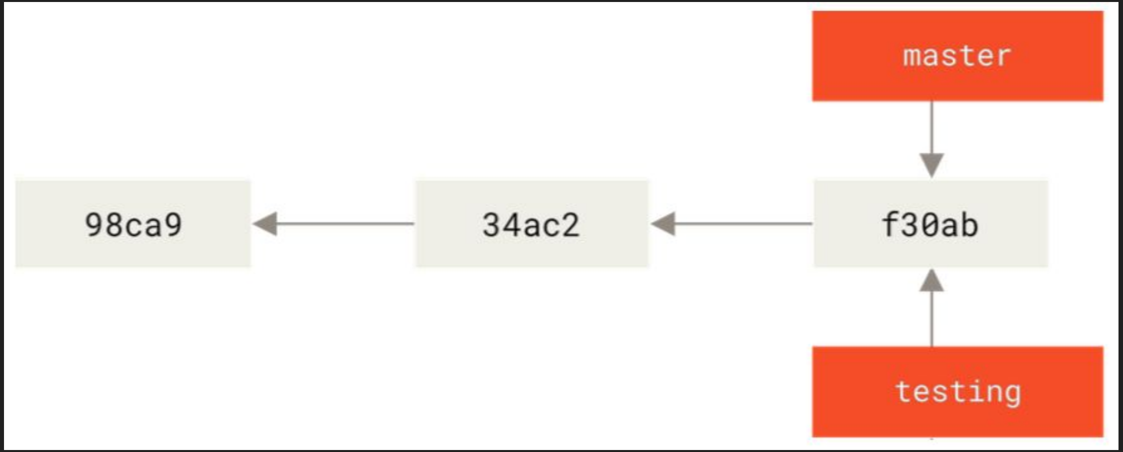
```
$ git config --global user.name "Name Surname"

$ git config --global user.email name.surname@studenti.unicam.it
```

4. You can check your settings at any time:

```
$git config −−list
```

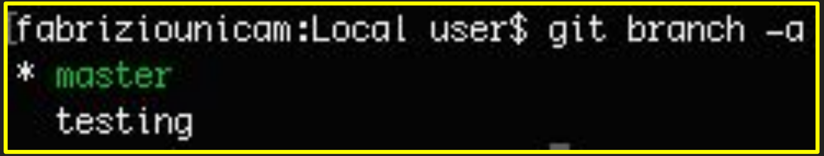
Creating a New Branch

1. Run git branch testing
2. Run git status



The git branch command only created a new branch — it didn't switch to that.

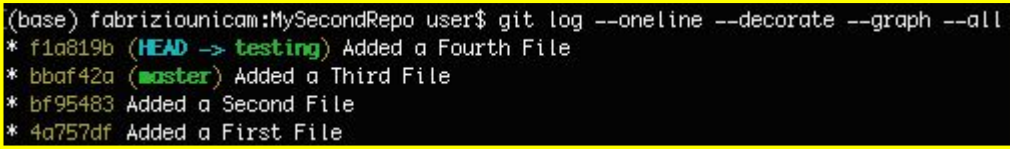
3. Run git branch -a



```
fabriziounicam:Local user$ git branch -a
* master
  testing
```

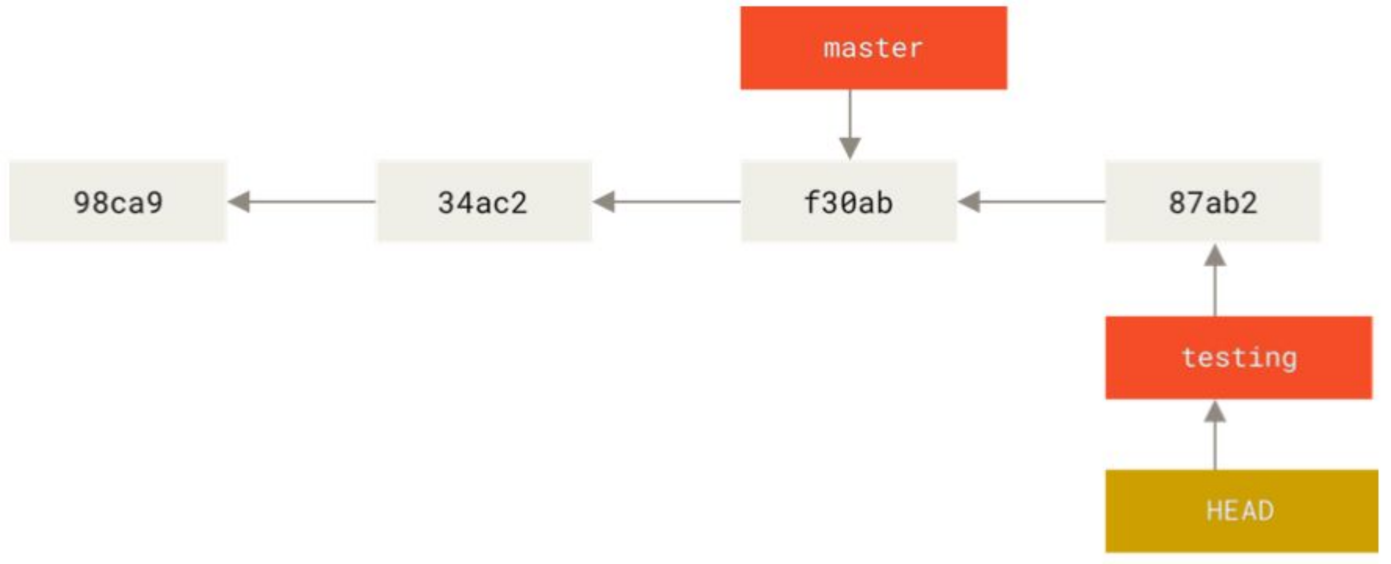
Commit to a New Branch

2. Run git checkout testing
3. Create a new file (or do some changes to the already available files)
4. Commit those changes
5. Run git log --oneline --decorate --graph --all



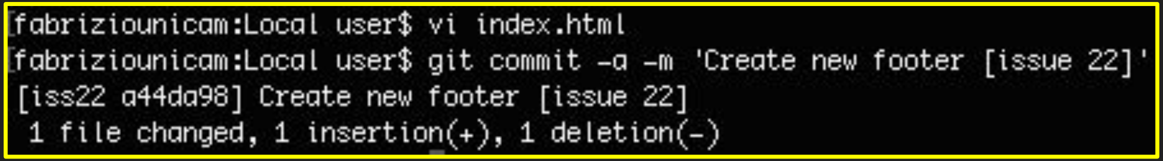
```
(base) fabriziounicam:MySecondRepo user$ git log --oneline --decorate --graph --all
* f1a819b (HEAD -> testing) Added a Fourth File
* bbaf42a (master) Added a Third File
* bf95483 Added a Second File
* 4a757df Added a First File
```

Your testing branch has moved forward



Branching and Merging

You do some changes and you commit



```
fabriziounicam:Local user$ vi index.html
fabriziounicam:Local user$ git commit -a -m 'Create new footer [issue 22]'
[iss22 a44da98] Create new footer [issue 22]
 1 file changed, 1 insertion(+), 1 deletion(-)
```

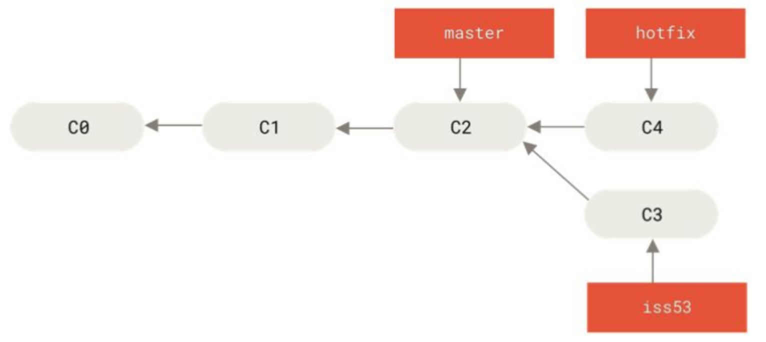
Now you get the call that there is an issue with the website, and you need to fix it immediately.

1. Run git checkout master
2. You have a hotfix to make. Let’s create a hotfix branch on which to work until it’s completed.
3. Run git checkout -b hotfix
4. Modify index.html file and commit the changes

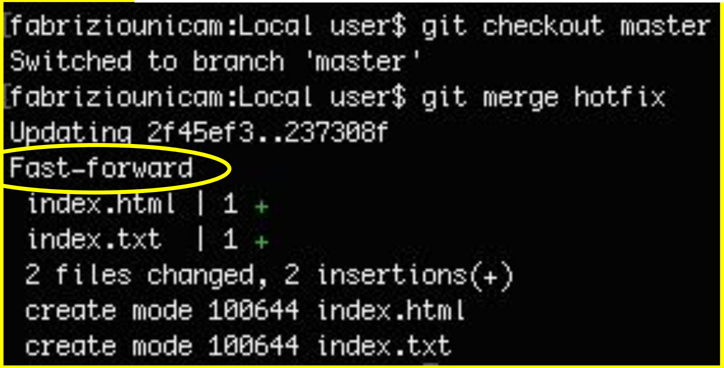
Branching and Merging

You can run your tests, make sure the hotfix is what you want, and finally merge the hotfix branch back into your master branch to deploy to production.

1. Run git checkout master
2. Run git merge hotfix



“Fast-forward” - when you try to merge one commit with a commit that can be reached by following the first commit’s history, Git simplifies things by moving the pointer forward because there is no divergent work to merge together



```
fabriziounicam:Local user$ git checkout master
Switched to branch 'master'
fabriziounicam:Local user$ git merge hotfix
Updating 2f45ef3..237308f
Fast-forward
 index.html | 1 +
 index.txt  | 1 +
 2 files changed, 2 insertions(+)
 create mode 100644 index.html
 create mode 100644 index.txt
```

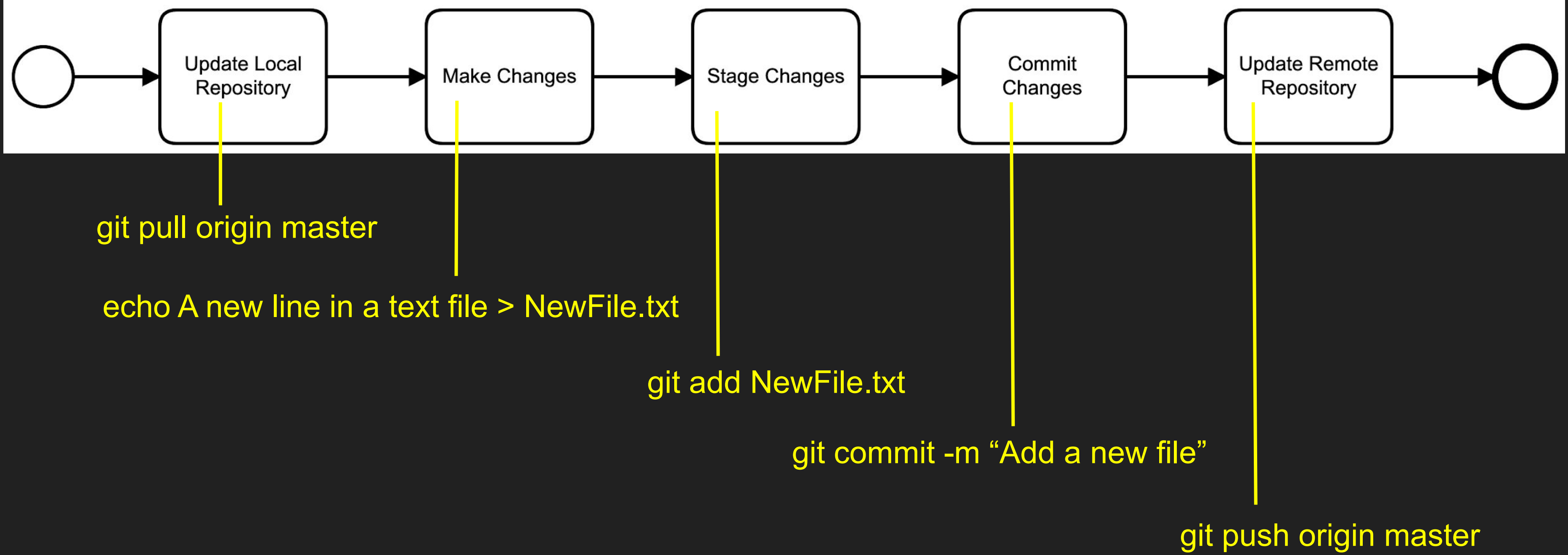
Lecture 2

A remote repository is a repository stored somewhere else.

Most programmers use hosting services like:

- **GitHub**,
- **BitBucket,**
- **GitLab**

Collaborative Workflow



Update Local Repository

Make Changes

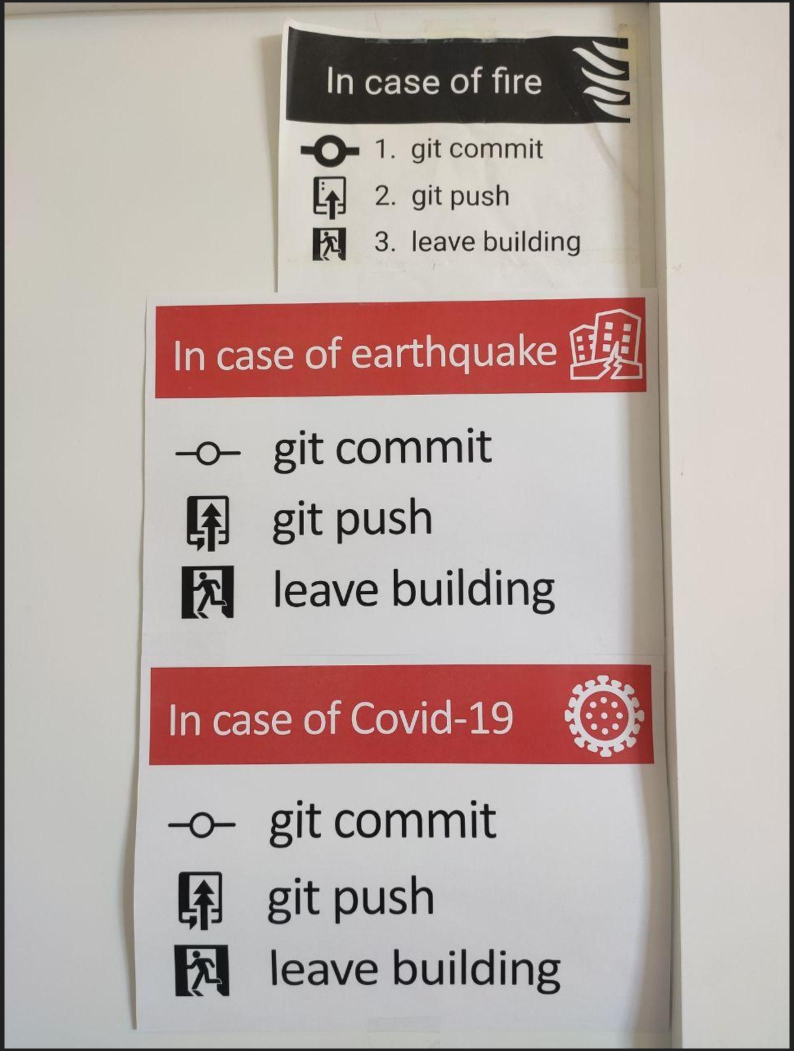
Stage Changes

Commit Changes

Update Remote Repository

```
git pull origin master

echo A new line in a text file > NewFile.txt

git add NewFile.txt

git commit -m “Add a new file”

git push origin master
```

In case of...



In case of fire

1. git commit
2. git push
3. leave building

In case of earthquake

git commit

git push

leave building

In case of Covid-19

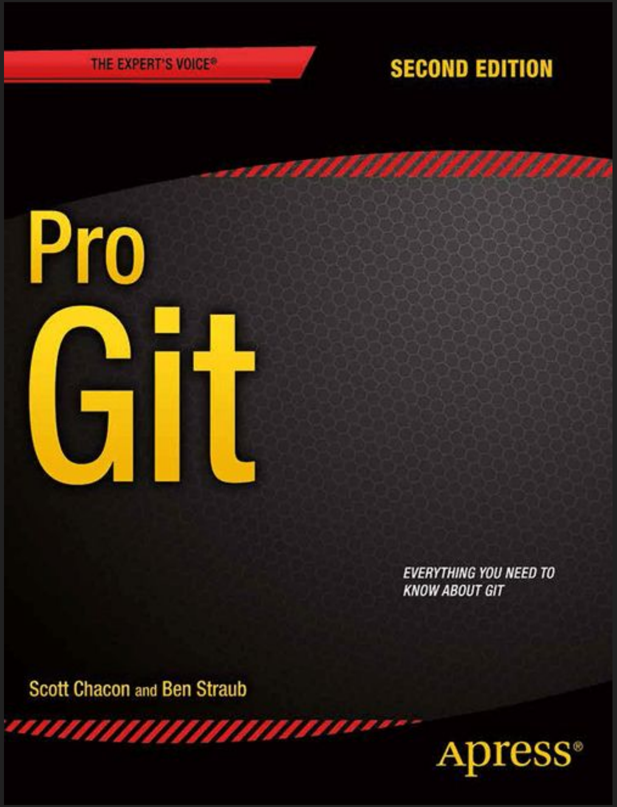
git commit

git push

leave building

Additional Materials

Pro Git
https://git-scm.com/book/en/v2
by Scott Chacon and Ben Straub


THE EXPERT'S VOICE®
SECOND EDITION
Pro Git
EVERYTHING YOU NEED TO KNOW ABOUT GIT
Scott Chacon and Ben Straub
Apress®

Lecture 4

Meeting the customers

1. Possibility to discuss with different customers for different projects (around 15min each)

2. A customer will tell you about his/her own needs

3. You are supposed to take notes and ask questions to get some clarification

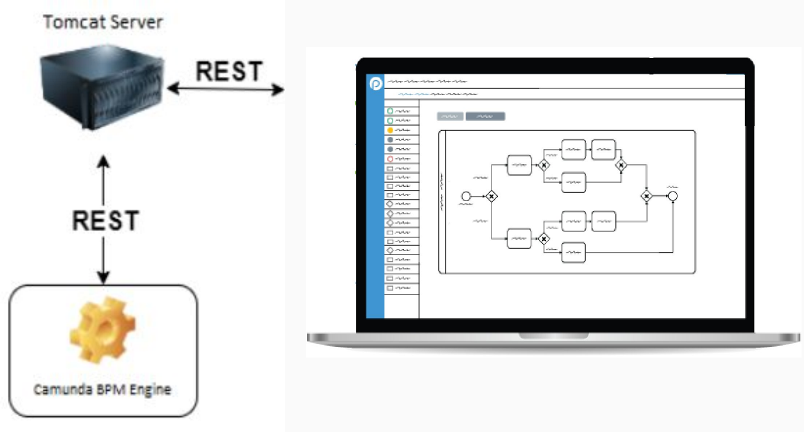
4. The notes that you take will help you in defining the user stories

NOTE: Customers are absolutely NOT AUTHORISED to give you additional information out of the lectures hours. So, do not bother them with questions, messages, mails or whatever comes to your mind.

IoT-Aware BPMN Platform

The project consists of implementing a **web application** that allows to **design and enact** IoT-Aware BPMN models

Group Acronym	Name	Surname	Role
MTV	Mattia	Romagnoli	
	Tommaso	Cippitelli	
	Vittorio	Rinaldi	



Customer: Ivan Compagnucci

IoT Platform

The project consists of realising an **IoT Platform** for managing **IoT devices.**

This platform must allow the **import**, **visualising** and **saving** of information related to IoT devices.

TLD	Tommaso	Carletti	
	Luca	Cervioni	
	Dmitry	Mingazov	Scrum Master
PSG	Francesco Pio	Stelluti	Scrum Master
	Marco	Zamponi	
	Luca	Fuligni	
	Michele	Russo	
SSU	Leonardo	Mogianesi	Scrum Master
	Luca	Tasso	
	Mattia	Giordani	
MWE	Gioele	Giachè	
	Lorenzo	Brancaleoni	
	Keerthi	Ravilla Subramanayam	

Customer: Arianna Fedeli

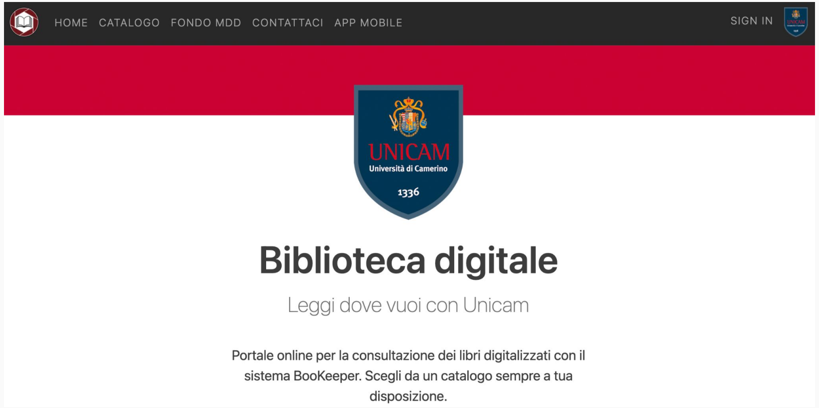
Digital Library

The project consists in developing a web/mobile application for accessing digital books. The system allows users to create a digital library and to read stored books, add notes bookmarks and share them with other users.

SMES	Shkemb	Abdullahu	
	Martin	Peraic	Scrum Master
	Eric Nuertey	Coleman	
	Sauro	Cesaretti	
YMLA	Matteo	Leonesi	Scrum Master
	Yuri	Paoloni	
	Luca	Fioravanti	
	Andrea	De Angelis	
DMD	Michele	Benedetti	
	Daniele	Moschini	Scrum Master
	Diego	Diomedi	

Customers: Federico Valeri, Melania Fattorini, Francesco Casoni

https://unicam.webex.com/meet/fabrizio.fornari



Biblioteca digitale

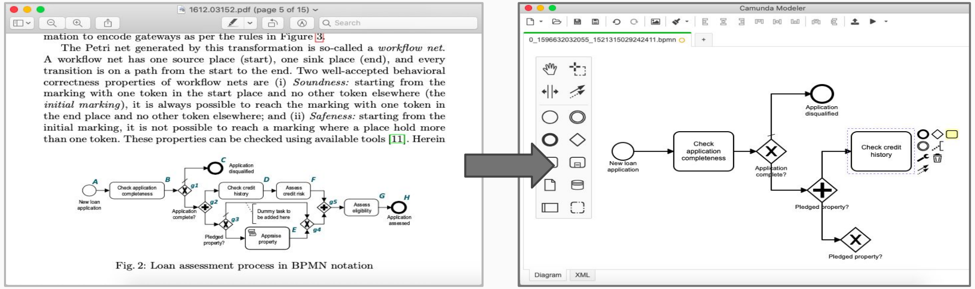
Leggi dove vuoi con Unicam

Portale online per la consultazione dei libri digitalizzati con il sistema BooKeeper. Scegli da un catalogo sempre a tua disposizione.

BPMN Redrawer

The project consists of implementing a web application that allows to upload images (.png) of BPMN models and turns those images in actual BPMN models stored in .bpmn format

RAM	Riccardo	Coltrinari	
	Alessandro	Antinori	
	Marco	Scarpetta	
FAB	Federico	Fabrizi	
	Alessandro	Zallocco	
	Bilel	Braiek	
MMB	Beatrice	Strappa	Team Member
	Massimiliano	Sampaolo	Team Member
	Matilde	Marcelletti	Scrum Master



mation to encode gateways as per the rules in Figure 3.

The Petri net generated by this transformation is so-called a *workflow net.* A workflow net has one source place (start), one sink place (end), and every transition is on a path from the start to the end. Two well-accepted behavioral correctness properties of workflow nets are (i) *Soundness:* starting from the marking with one token in the start place and no other token elsewhere (the *initial marking*), it is always possible to reach the marking with one token in the end place and no other token elsewhere; and (ii) *Safeness:* starting from the initial marking, it is not possible to reach a marking where a place hold more than one token. These properties can be checked using available tools [11]. Herein

Fig. 2: Loan assessment process in BPMN notation

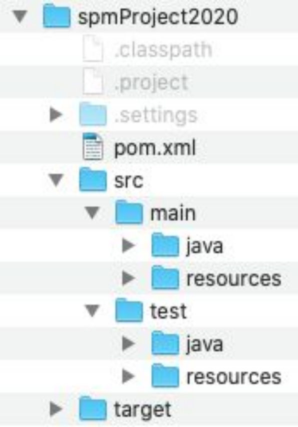
Customer: Fabrizio Fornari

Lecture 5

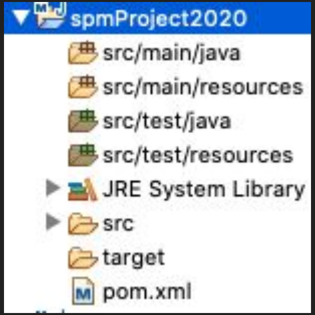
Apache Maven is an open source, standards-based project management framework that simplifies the building, testing, reporting, and packaging of projects.

http://maven.apache.org

Maven - Convention over Configuration



```
▼ spmProject2020
      .classpath
      .project
    ▶ .settings
      pom.xml
    ▼ src
        ▼ main
            ▶ java
            ▶ resources
        ▼ test
            ▶ java
            ▶ resources
    ▶ target
```



```
▼ spmProject2020
      src/main/java
      src/main/resources
      src/test/java
      src/test/resources
    ▶ JRE System Library
    ▶ src
      target
      pom.xml
```

- **spmProject2020** is the root folder of the project. Typically, the name of the root folder matches the name of the generated artifact.

- **src** contains project-related artifacts such as source code or property files, which you typically would like to manage in a source control management (SCM) system, such as Git.

- **src/main/java** folder contains the Java source code.

- **src/test/java** folder contains the Java unit test code.

- **target** folder holds generated artifacts, such as .class files. Generated artifacts are typically not stored in SCM, so you don’t commit the target folder and its contents into SCM.

- **pom.xml** file. It holds project and configuration information, such as dependencies and plug-ins

Maven - POM

Maven project structure and contents are declared in an xml file, pom.xml, referred as Project Object Model (POM), which is the fundamental unit of the entire Maven system.

The POM contains information about the project and various configuration details used by Maven to build the project(s).

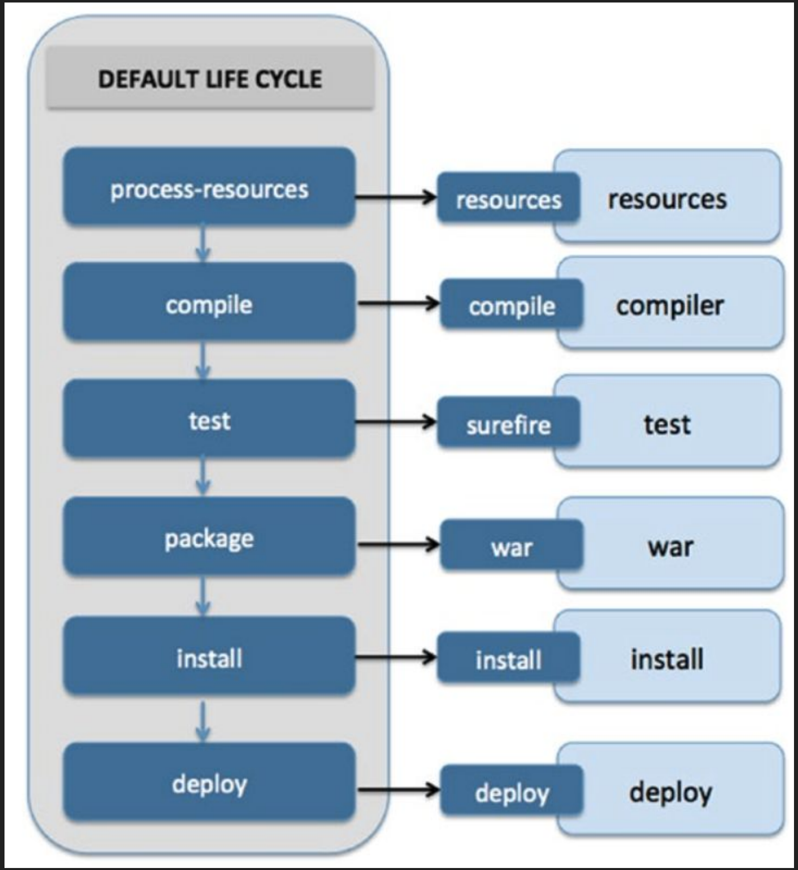
POM also contains the goals and plugins. While executing a task or goal, Maven looks for the POM in the current directory. It reads the POM, gets the needed configuration information, and then executes the goal.

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

- project dependencies
- plugins
- goals
- build profiles
- project version

Maven Lifecycle

validate	Runs checks to ensure that the project is correct and that all dependencies are downloaded and available.
compile	Compiles the source code.
test	Runs unit tests using frameworks. This step doesn’t require that the application be packaged.
package	Assembles compiled code into a distributable format, such as JAR or WAR.
install	Installs the packaged archive into a local repository. The archive is now available for use by any project running on that machine.
deploy	Pushes the built archive into a remote repository for use by other teams and team members.



Maven - Dependency Management

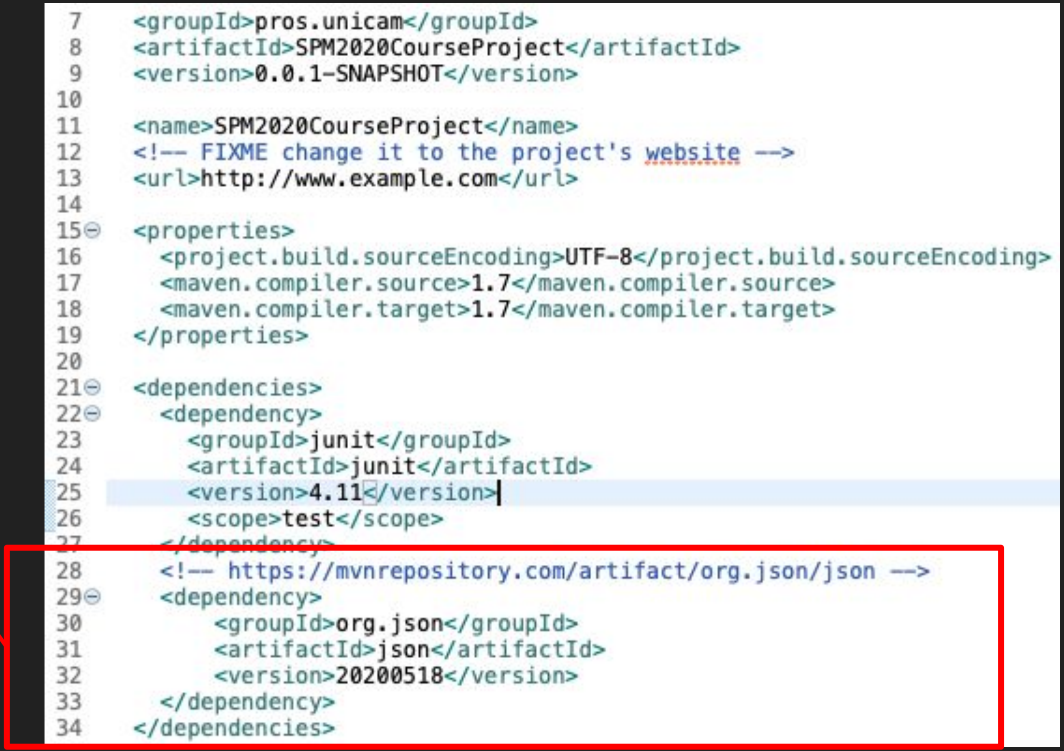
MVNREPOSITORY

https://mvnrepository.com/

Search the library you need and add it to the POM

I searched for a JSON library

I added it to the POM and I build the project



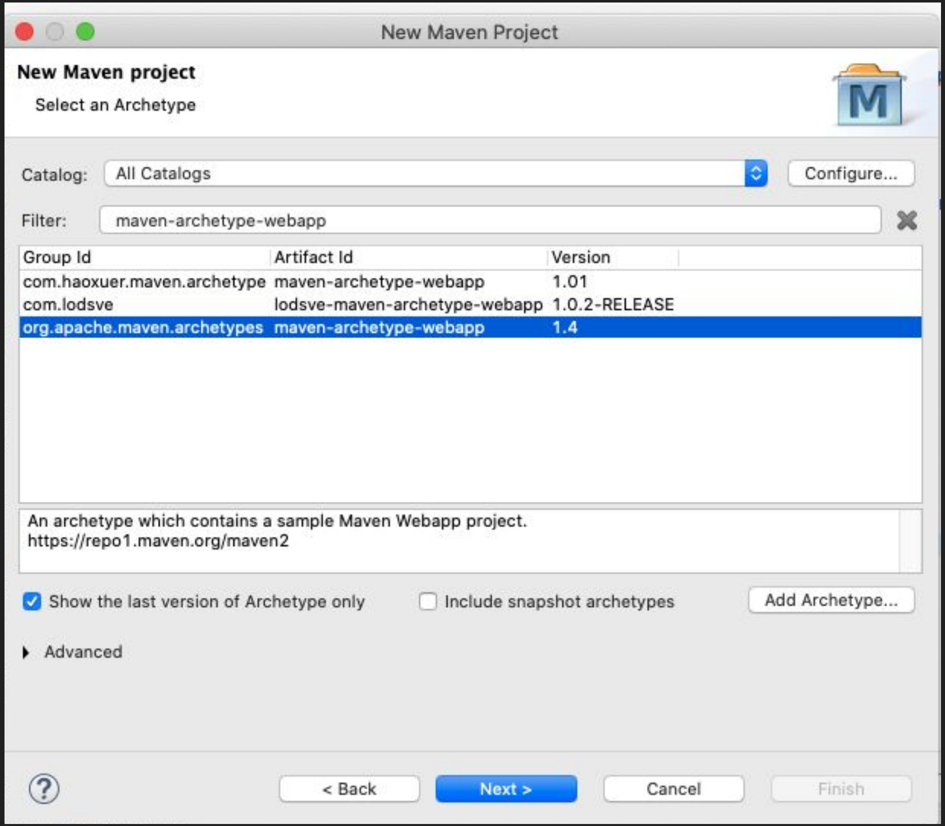
```
  <groupId>pros.unicam</groupId>
  <artifactId>SPM2020CourseProject</artifactId>
  <version>0.0.1-SNAPSHOT</version>

  <name>SPM2020CourseProject</name>
  <!-- FIXME change it to the project's website -->
  <url>http://www.example.com</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.7</maven.compiler.source>
    <maven.compiler.target>1.7</maven.compiler.target>
  </properties>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>
    <!-- https://mvnrepository.com/artifact/org.json/json -->
    <dependency>
        <groupId>org.json</groupId>
        <artifactId>json</artifactId>
        <version>20200518</version>
    </dependency>
  </dependencies>
```

Maven - Archetypes

Maven archetypes are project templates that allow users to generate new projects easily

Create a Maven Project by following: File → New → Other → Maven Project → Next

Insert “maven-archetype- webapp”, select and proceed



Catalog: All Catalogs

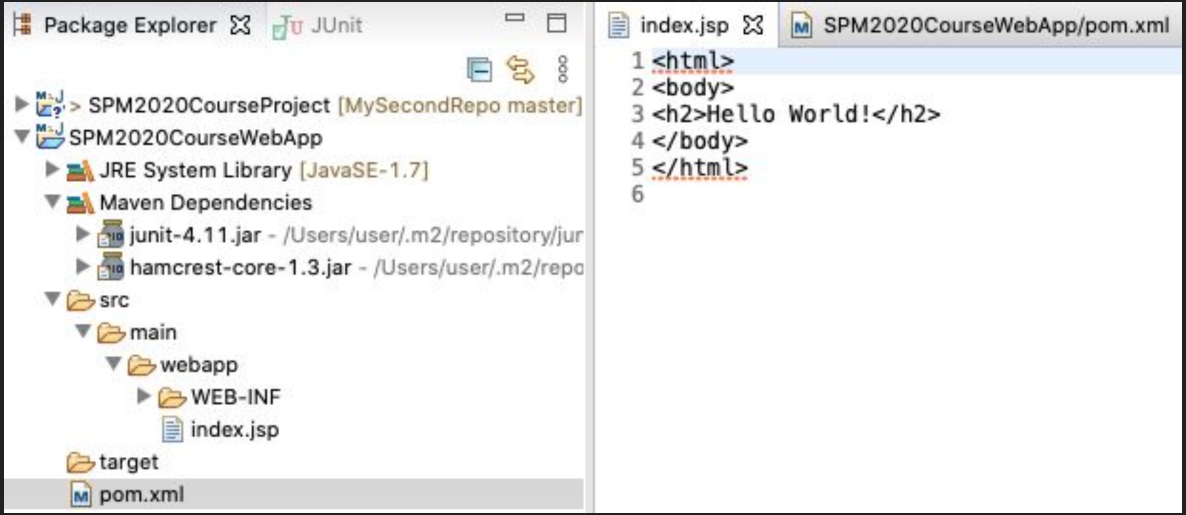
Filter: maven-archetype-webapp

Group Id	Artifact Id	Version
com.haoxuer.maven.archetype	maven-archetype-webapp	1.01
com.lodsve	lodsve-maven-archetype-webapp	1.0.2-RELEASE
org.apache.maven.archetypes	maven-archetype-webapp	1.4

An archetype which contains a sample Maven Webapp project.
https://repo1.maven.org/maven2

☑ Show the last version of Archetype only ☐ Include snapshot archetypes

Maven - Archetype WebApp

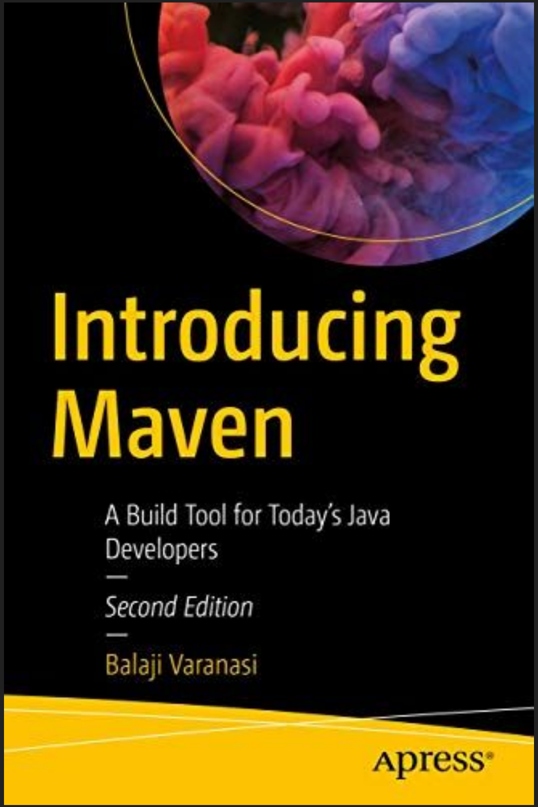

Package Explorer
JUnit
> SPM2020CourseProject [MySecondRepo master]
SPM2020CourseWebApp
JRE System Library [JavaSE-1.7]
Maven Dependencies
junit-4.11.jar - /Users/user/.m2/repository/jur
hamcrest-core-1.3.jar - /Users/user/.m2/repo
src
main
webapp
WEB-INF
index.jsp
target
pom.xml
index.jsp
SPM2020CourseWebApp/pom.xml

```
<html>
<body>
<h2>Hello World!</h2>
</body>
</html>

```

Maven - Additional Material

Introducing Maven:
A Build Tool for Today’s Java Developers.

by Balaji Varanasi



Apache Tomcat

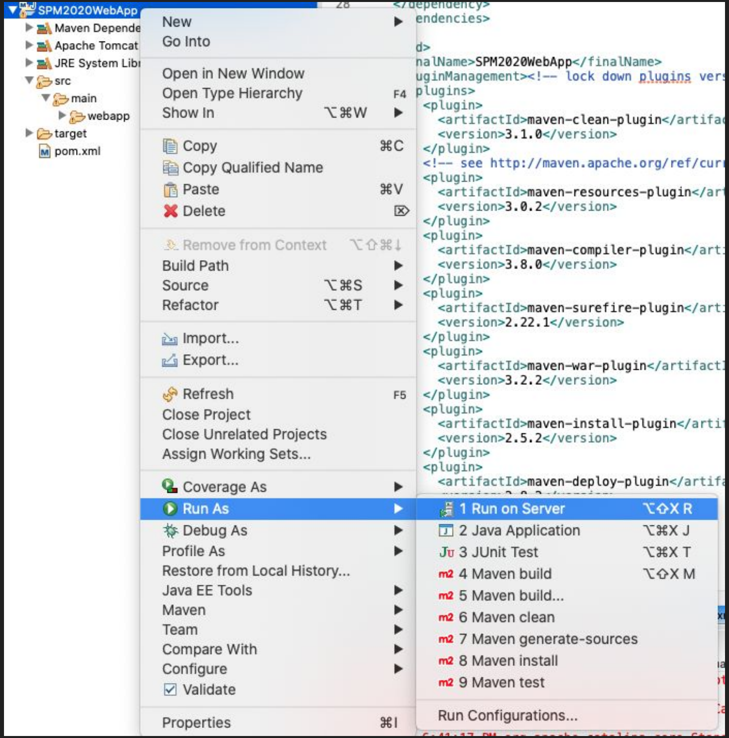
The Apache Tomcat® software is an open source implementation of the Java Servlet, JavaServer Pages, Java Expression Language and Java WebSocket technologies.

http://tomcat.apache.org/

Download Tomcat
https://tomcat.apache.org/download-90.cgi

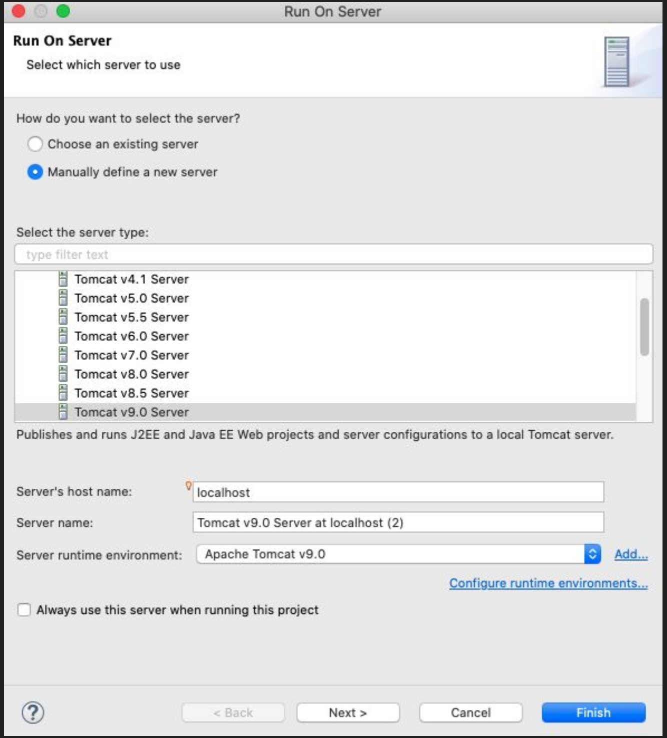
Run Your Application

Run On Server



```
  </dependency>
endencies>

d>
nalName>SPM2020WebApp</finalName>
uginManagement><!-- lock down plugins vers
plugins>
 <plugin>
   <artifactId>maven-clean-plugin</artifa
   <version>3.1.0</version>
 </plugin>
 <!-- see http://maven.apache.org/ref/cur
 <plugin>
   <artifactId>maven-resources-plugin</ar
   <version>3.0.2</version>
 </plugin>
 <plugin>
   <artifactId>maven-compiler-plugin</art
   <version>3.8.0</version>
 </plugin>
 <plugin>
   <artifactId>maven-surefire-plugin</art
   <version>2.22.1</version>
 </plugin>
 <plugin>
   <artifactId>maven-war-plugin</artifact
   <version>3.2.2</version>
 </plugin>
 <plugin>
   <artifactId>maven-install-plugin</arti
   <version>2.5.2</version>
 </plugin>
 <plugin>
   <artifactId>maven-deploy-plugin</artif
```

Pick the version you installed



Run On Server

Run On Server

Select which server to use

How do you want to select the server?

○ Choose an existing server

● Manually define a new server

Select the server type:

type filter text

- Tomcat v4.1 Server
- Tomcat v5.0 Server
- Tomcat v5.5 Server
- Tomcat v6.0 Server
- Tomcat v7.0 Server
- Tomcat v8.0 Server
- Tomcat v8.5 Server
- Tomcat v9.0 Server

Publishes and runs J2EE and Java EE Web projects and server configurations to a local Tomcat server.

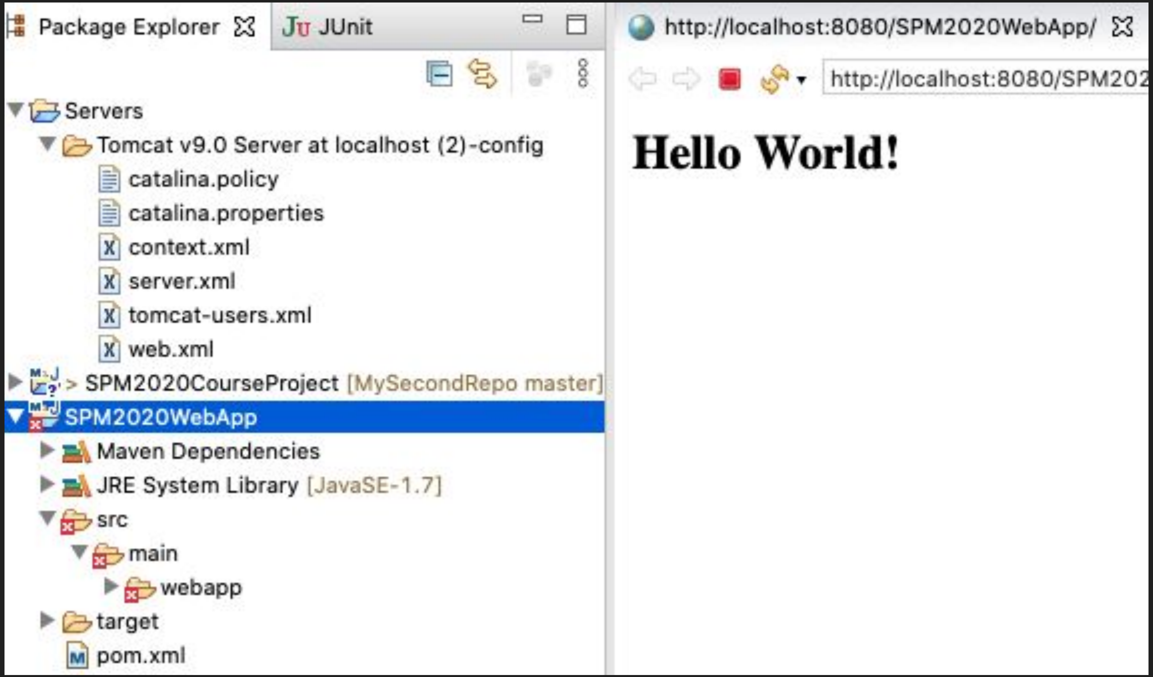
Server's host name: localhost

Server name: Tomcat v9.0 Server at localhost (2)

Server runtime environment: Apache Tomcat v9.0

□ Always use this server when running this project

Run Your Application



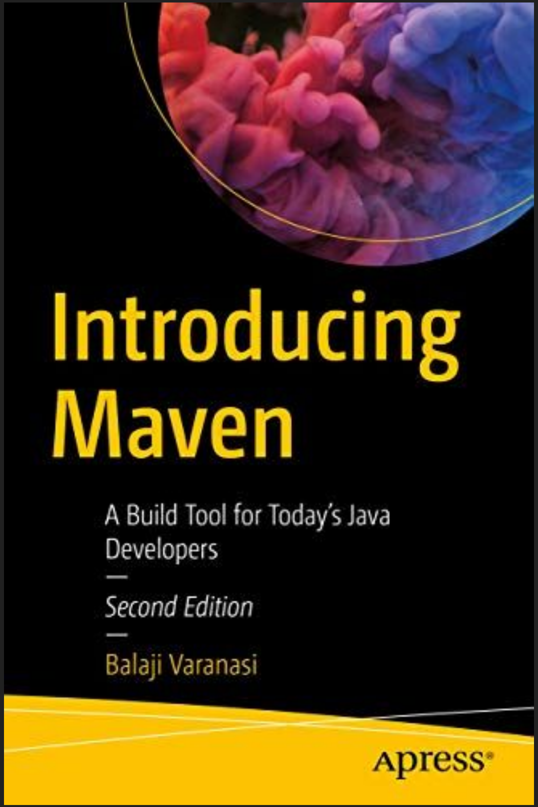
Package Explorer
JUnit
Servers
Tomcat v9.0 Server at localhost (2)-config
catalina.policy
catalina.properties
context.xml
server.xml
tomcat-users.xml
web.xml
> SPM2020CourseProject [MySecondRepo master]
SPM2020WebApp
Maven Dependencies
JRE System Library [JavaSE-1.7]
src
main
webapp
target
pom.xml
http://localhost:8080/SPM2020WebApp/
http://localhost:8080/SPM202

Hello World!

Maven - Additional Material

Introducing Maven:
A Build Tool for Today's Java Developers.

by Balaji Varanasi



Introducing Maven

A Build Tool for Today's Java Developers

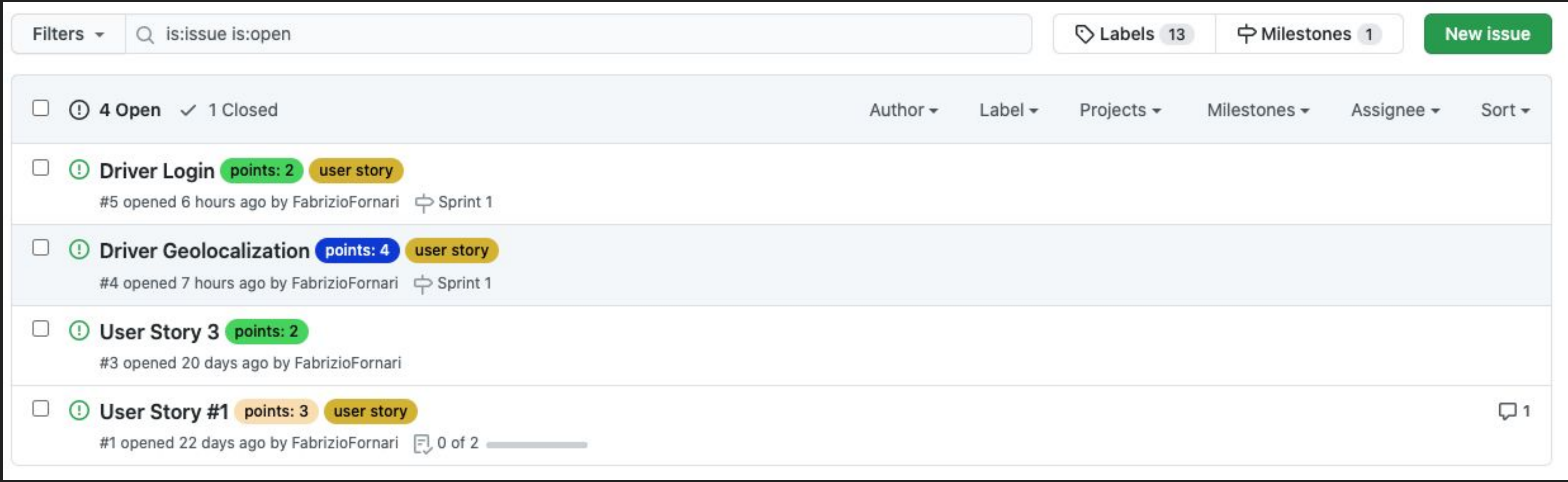
Second Edition

Balaji Varanasi

Apress®

Github Project Settings

https://github.com/FabrizioFornari/SPM2020Template



□ 4 Open ✓ 1 Closed

□ Driver Login points: 2 user story
#5 opened 6 hours ago by FabrizioFornari Sprint 1

□ Driver Geolocalization points: 4 user story
#4 opened 7 hours ago by FabrizioFornari Sprint 1

□ User Story 3 points: 2
#3 opened 20 days ago by FabrizioFornari

□ User Story #1 points: 3 user story 1
#1 opened 22 days ago by FabrizioFornari 0 of 2

Lecture 6 (Sprint Meeting)

A User Story is a simple and quick description of a specific way that the user will use the software. Generally between one and four sentences long.

Can generally follow a template:

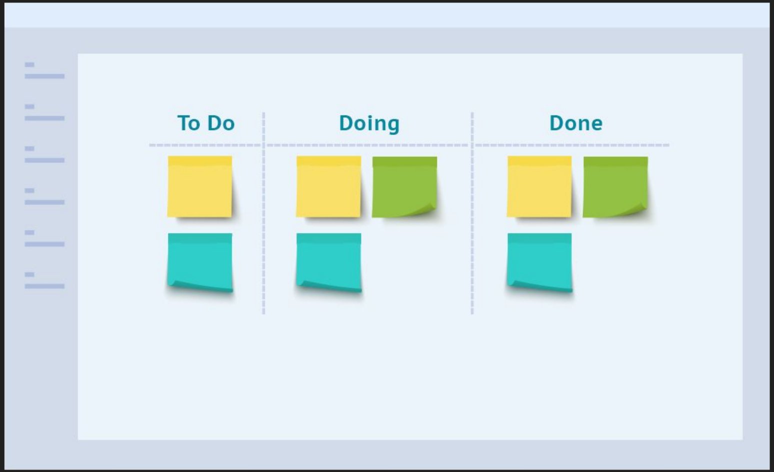
As a <*type of user*>,
I want to <*specific action I'm taking*>
so that <*what I want to happen as a result*>.

e.g. "As a customer, I want to be able to create an account so that I can see the purchases I made in the last year to help me budget for next year."

Assign a value to estimate the effort needed to elaborate a user story (e.g., 1 to 5).

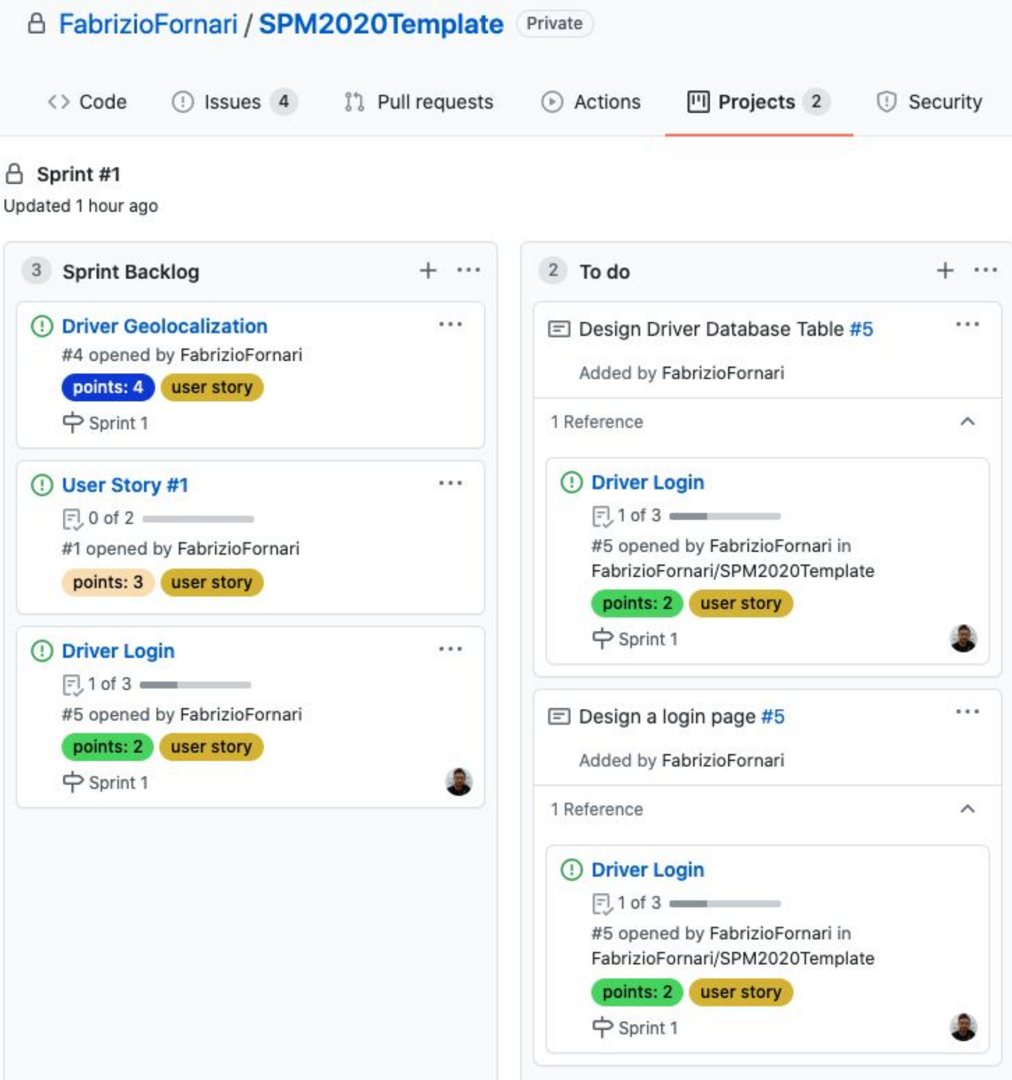
Kanban

Kanban is a visual system for managing work as it moves through a process. Kanban visualizes both the process (the workflow) and the actual work passing through that process.



Kanban, also spelt “kamban” in Japanese, translates to “Billboard” (“signboard” in Chinese) that indicates “available capacity (to work)”. Kanban is a concept related to lean and just-in-time (JIT) production, where it is used as a scheduling system that tells you what to produce, when to produce it, and how much to produce.

Divide User Stories Into Small Tasks



FabrizioFornari / SPM2020Template Private

Sprint #1
Updated 1 hour ago

3 Sprint Backlog

Driver Geolocalization
#4 opened by FabrizioFornari
points: 4 user story
Sprint 1

User Story #1
0 of 2
#1 opened by FabrizioFornari
points: 3 user story

Driver Login
1 of 3
#5 opened by FabrizioFornari
points: 2 user story
Sprint 1

2 To do

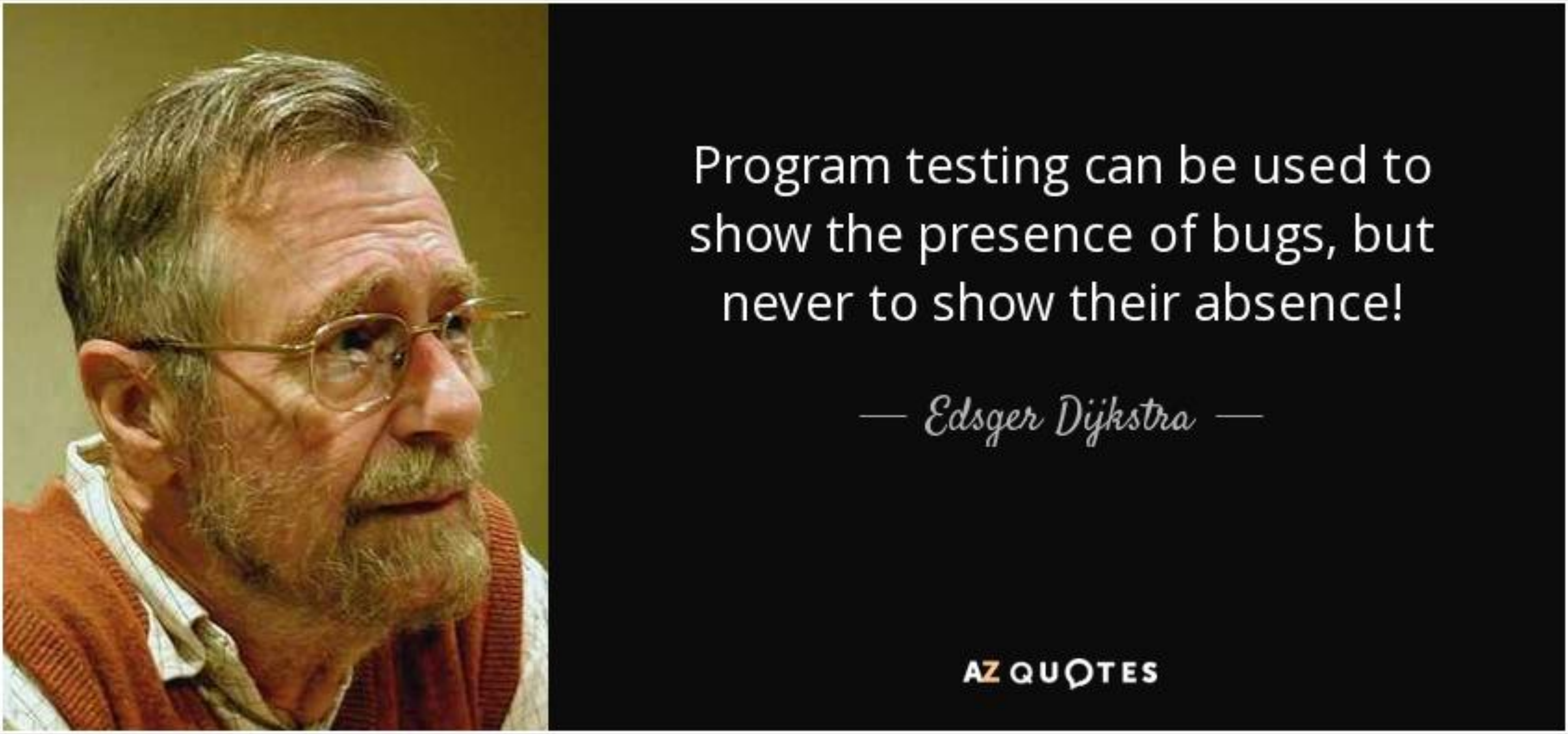
Design Driver Database Table #5
Added by FabrizioFornari
1 Reference
Driver Login
1 of 3
#5 opened by FabrizioFornari in FabrizioFornari/SPM2020Template
points: 2 user story
Sprint 1

Design a login page #5
Added by FabrizioFornari
1 Reference
Driver Login
1 of 3
#5 opened by FabrizioFornari in FabrizioFornari/SPM2020Template
points: 2 user story
Sprint 1

User Stories vs Tasks
www.mountaingoatsoftware.com

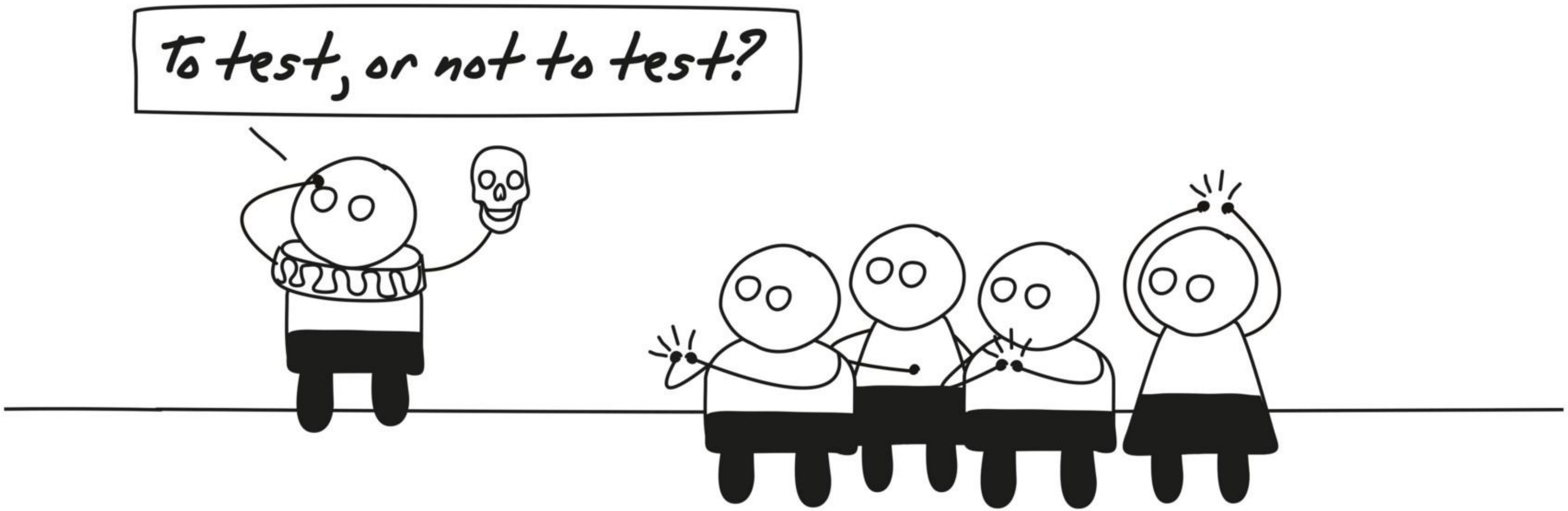
Lecture 7

Testing is the activity of finding out whether a piece of code (a method, class, or program) produces the intended behavior.



Program testing can be used to show the presence of bugs, but never to show their absence!

— Edsger Dijkstra —

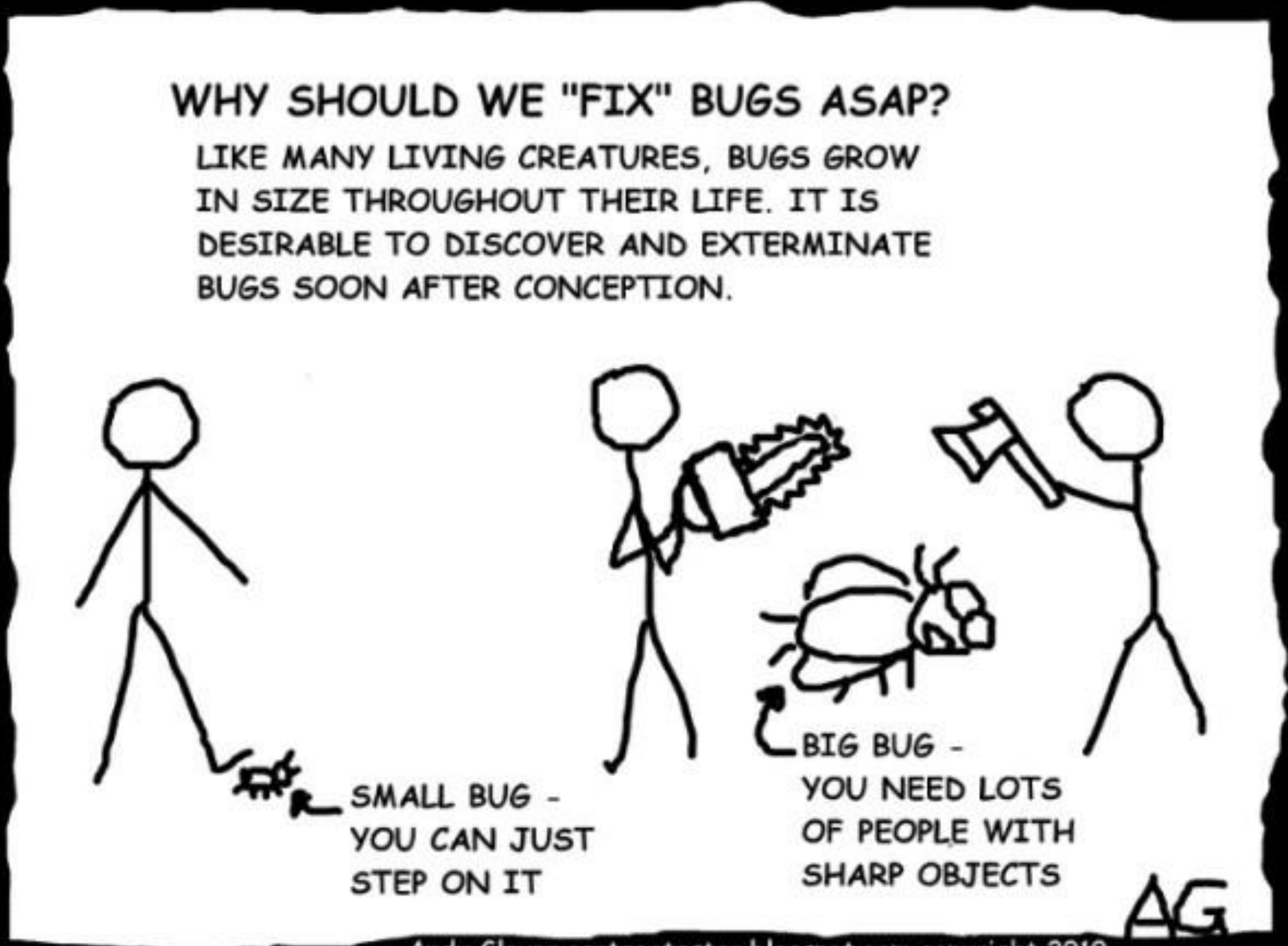

To test, or not to test?

Testing

The purpose of testing is to find bugs and errors.

Debugging

The purpose of debugging is to correct those bugs found during testing.


WHY SHOULD WE "FIX" BUGS ASAP?
LIKE MANY LIVING CREATURES, BUGS GROW IN SIZE THROUGHOUT THEIR LIFE. IT IS DESIRABLE TO DISCOVER AND EXTERMINATE BUGS SOON AFTER CONCEPTION.
SMALL BUG - YOU CAN JUST STEP ON IT
BIG BUG - YOU NEED LOTS OF PEOPLE WITH SHARP OBJECTS
AG

Andy Glover cartoontester.blogspot.com copyright 2010

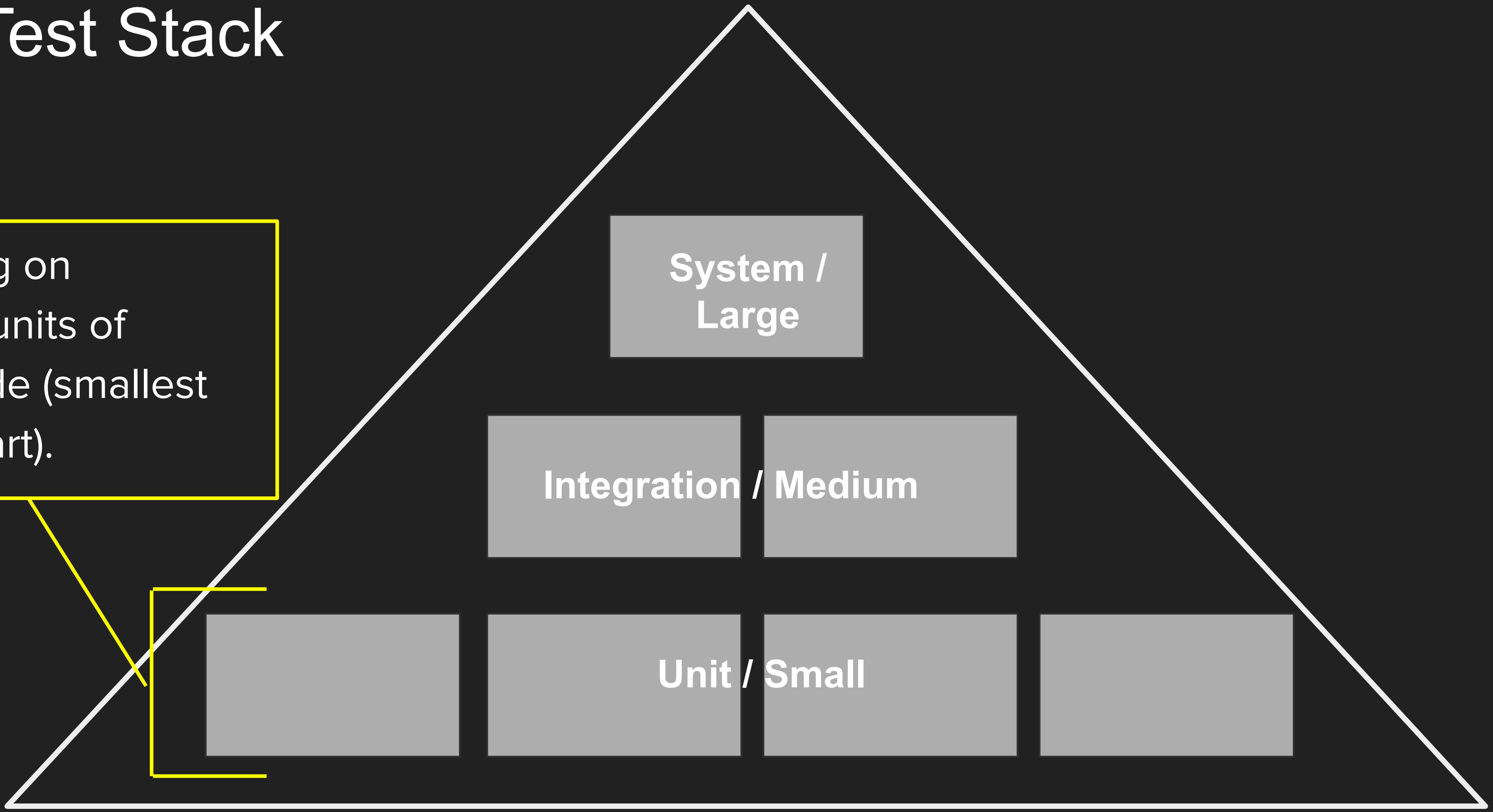
Test Sizes

Size & Time

Feature	Small	Medium	Large
Network access	No	localhost only	Yes
Database	No	Yes	Yes
File system access	No	Yes	Yes
Use external systems	No	Discouraged	Yes
Multiple threads	No	Yes	Yes
Sleep statements	No	Yes	Yes
System properties	No	Yes	Yes
Time limit (seconds)	60	300	900+

Google Testing Blog ‘Test Sizes’

The Test Stack



Unit testing on individual units of source code (smallest testable part).

JUnit

- JUnit (http://junit.org/) is a test framework which uses annotations to identify methods that specify a test. Typically these test methods are contained in a class which is only used for testing. It is typically called a *Test class*.

- Current version is JUnit 5

JUnit test example - MyClassTest

```
package test;

import static org.junit.jupiter.api.Assertions.assertEquals;
import org.junit.jupiter.api.Test;
import main.MyClass;

public class MyClassTest {

  @Test
  public void testMultiply() {
    MyClass tester = new MyClass();
    assertEquals(50, tester.multiply(10, 5),"10 x 5 must be 50");
  }
}
```

I know I should test, but everything?

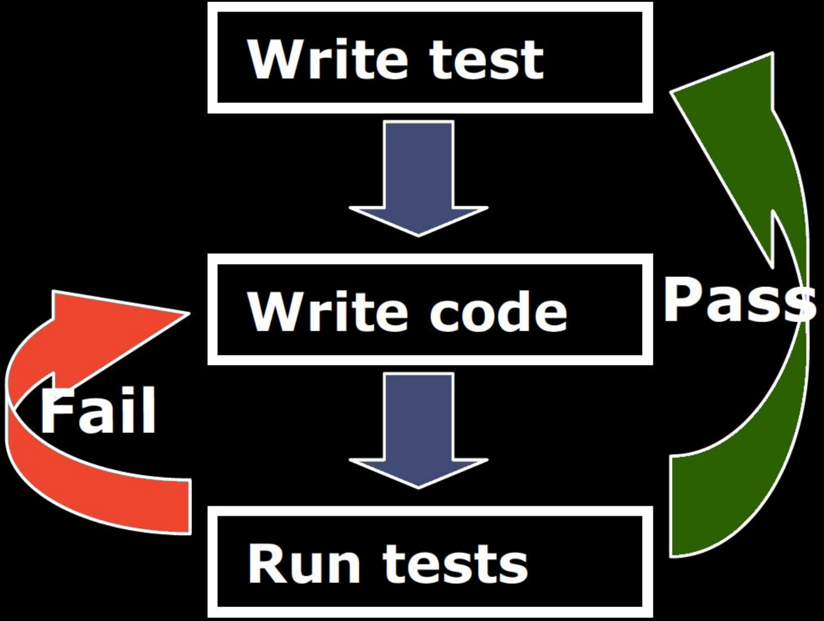
Best practices

- Tests should be written before the code (TDD - Test driven development)
- Test everything that could reasonably break.
- If it can’t break on its own, it’s too simple to break (like most get and set methods).
- Run all your unit tests as often as possible

TDD cycle

- Proceeds step by step
  a. Write a test.
  b. Design and implement just enough to make the test pass.
  c. Repeat.

- Testing and coding alternate in very small steps
  - Duration of one cycle should be a few minutes
  - Small steps – difficult to make mistake



Write test

Write code

Run tests

Fail

Pass

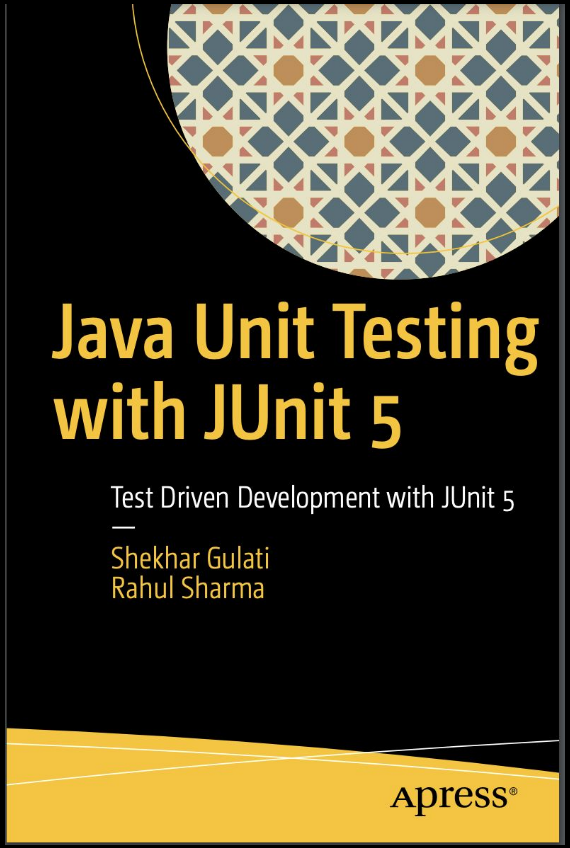
Additional Material

Check this out:
JUnit 5 User Guide

https://junit.org/junit5/docs/current/user-guide/index.pdf

or

https://junit.org/junit5/docs/current/user-guide/



Java Unit Testing with JUnit 5

Test Driven Development with JUnit 5

—

Shekhar Gulati
Rahul Sharma

Apress®

Lecture 8

Today:

- 10 groups
- 10 minutes of discussion with each group

The objectives:

- Checking the Team Status
- Checking the User Stories definition
- Asking and Answering Questions

Lecture 9

- Unit Testing
- Integration Testing
- Regression Testing
- ...

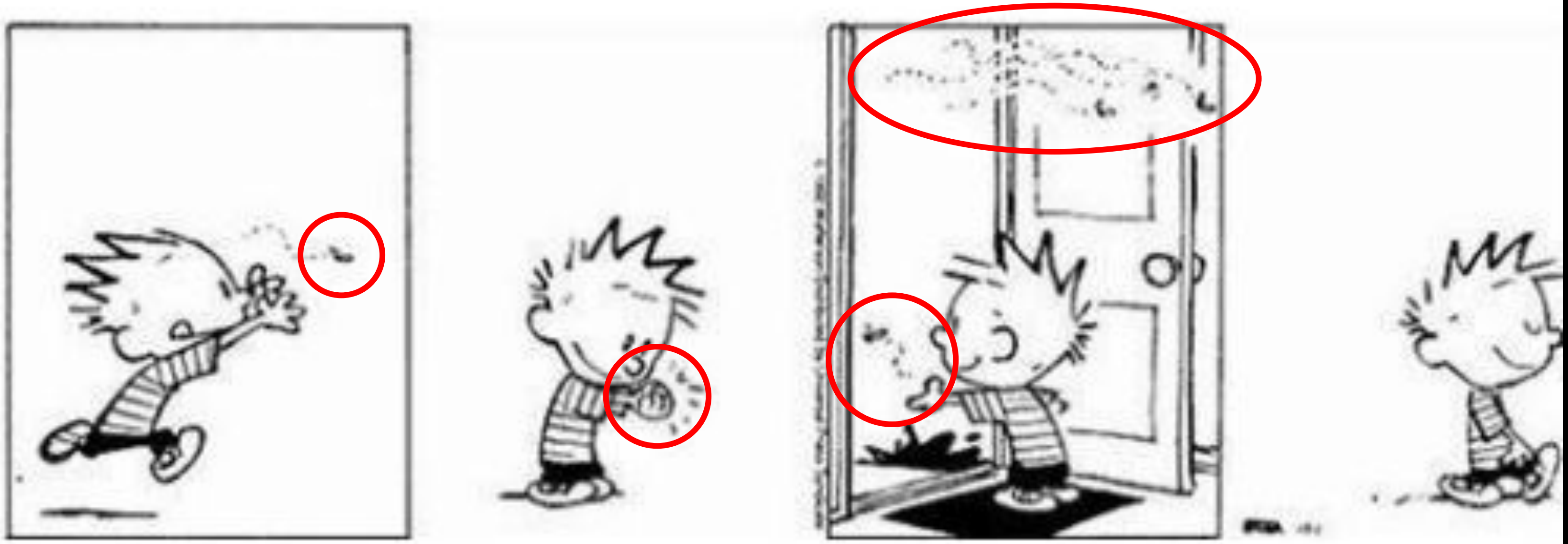
https://www.softwaretestinghelp.com/types-of-software-testing/

Integration Testing

Individual modules are combined and tested as a group.
Data transfer between the modules is tested as well.

Regression:

"when you fix one bug, you introduce several newer bugs."



Regression Testing

Test cases are re-executed in order to check whether the previous functionality of the application is working fine and the new changes have not introduced any new bugs.

This test can be performed on a new build when there is a significant change in the original functionality, even in correspondence of a single bug fix.

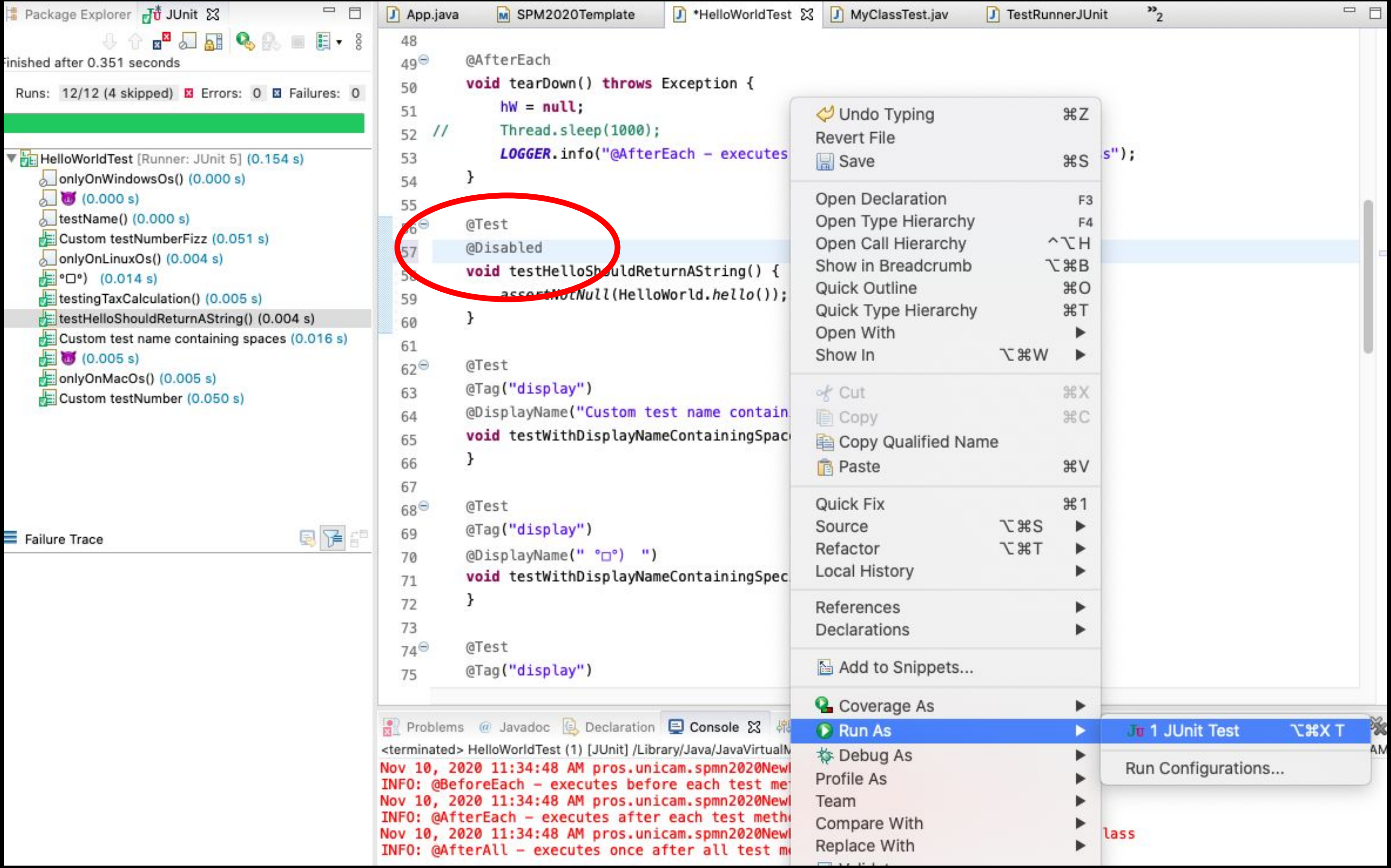
Manual Testing

The oldest type of software testing.

It requires a tester to perform manual test operations on the test software without automation scripts.

The tester choose which tests to run, when to run them, and how many times.

Manual Testing in Eclipse



Finished after 0.351 seconds

Runs: 12/12 (4 skipped) Errors: 0 Failures: 0

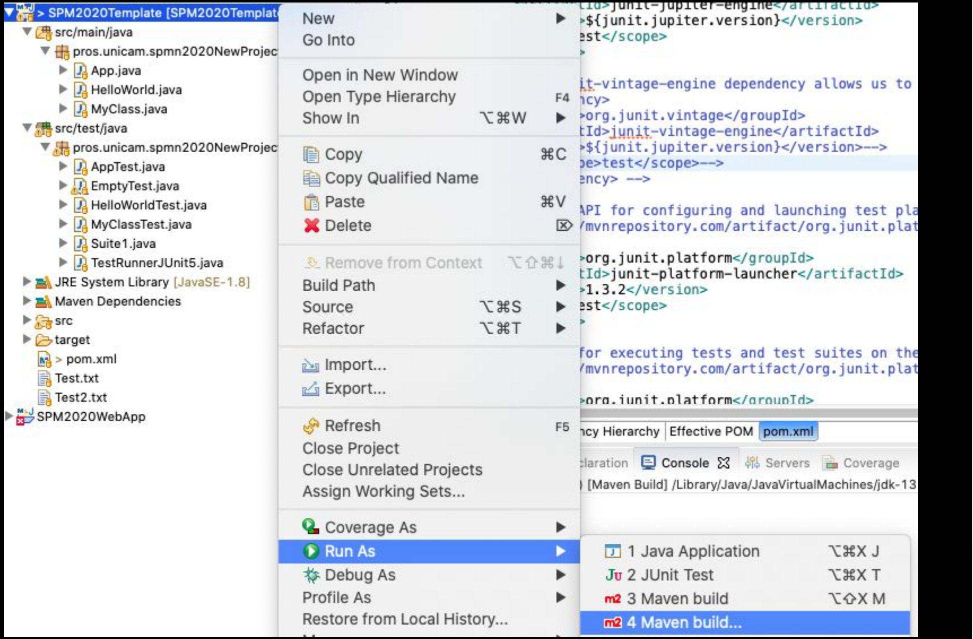
- HelloWorldTest [Runner: JUnit 5] (0.154 s)
  - onlyOnWindowsOs() (0.000 s)
  - 😈 (0.000 s)
  - testName() (0.000 s)
  - Custom testNumberFizz (0.051 s)
  - onlyOnLinuxOs() (0.004 s)
  - °□°) (0.014 s)
  - testingTaxCalculation() (0.005 s)
  - testHelloShouldReturnAString() (0.004 s)
  - Custom test name containing spaces (0.016 s)
  - 😈 (0.005 s)
  - onlyOnMacOs() (0.005 s)
  - Custom testNumber (0.050 s)

```
    @AfterEach
    void tearDown() throws Exception {
        hW = null;
//      Thread.sleep(1000);
        LOGGER.info("@AfterEach - executes                                    s");
    }

    @Test
    @Disabled
    void testHelloShouldReturnAString() {
        assertNotNull(HelloWorld.hello());
    }

    @Test
    @Tag("display")
    @DisplayName("Custom test name contain
    void testWithDisplayNameContainingSpac
    }

    @Test
    @Tag("display")
    @DisplayName(" °□°)  ")
    void testWithDisplayNameContainingSpec
    }

    @Test
    @Tag("display")
```

```
<terminated> HelloWorldTest (1) [JUnit] /Library/Java/JavaVirtualM
Nov 10, 2020 11:34:48 AM pros.unicam.spmn2020NewI
INFO: @BeforeEach - executes before each test me
Nov 10, 2020 11:34:48 AM pros.unicam.spmn2020NewI
INFO: @AfterEach - executes after each test meth
Nov 10, 2020 11:34:48 AM pros.unicam.spmn2020NewI                              lass
INFO: @AfterAll - executes once after all test me
```

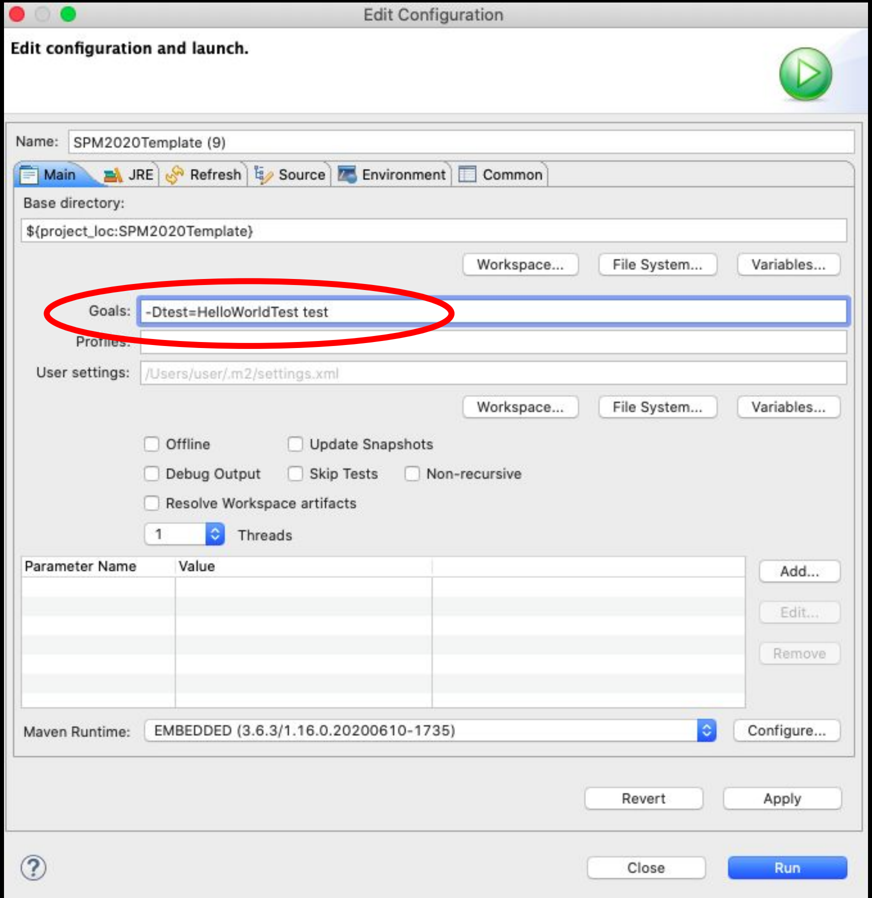
Manual Testing with Maven

- Run a single test class:
  -Dtest=<NameOfTheTestClass> test



```
tId>junit-jupiter-engine</artifactId>
>${junit.jupiter.version}</version>
est</scope>
>

it-vintage-engine dependency allows us to
ncy>
>org.junit.vintage</groupId>
tId>junit-vintage-engine</artifactId>
>${junit.jupiter.version}</version>-->
pe>test</scope>-->
ency> -->

API for configuring and launching test pla
/mvnrepository.com/artifact/org.junit.plat

>org.junit.platform</groupId>
tId>junit-platform-launcher</artifactId>
>1.3.2</version>
est</scope>
>

for executing tests and test suites on the
/mvnrepository.com/artifact/org.junit.plat

>org.junit.platform</groupId>
```

) [Maven Build] /Library/Java/JavaVirtualMachines/jdk-13



Edit Configuration

Edit configuration and launch.

Name: SPM2020Template (9)

Base directory: ${project_loc:SPM2020Template}

Goals: -Dtest=HelloWorldTest test

Profiles:

User settings: /Users/user/.m2/settings.xml

□ Offline □ Update Snapshots

□ Debug Output □ Skip Tests □ Non-recursive

□ Resolve Workspace artifacts

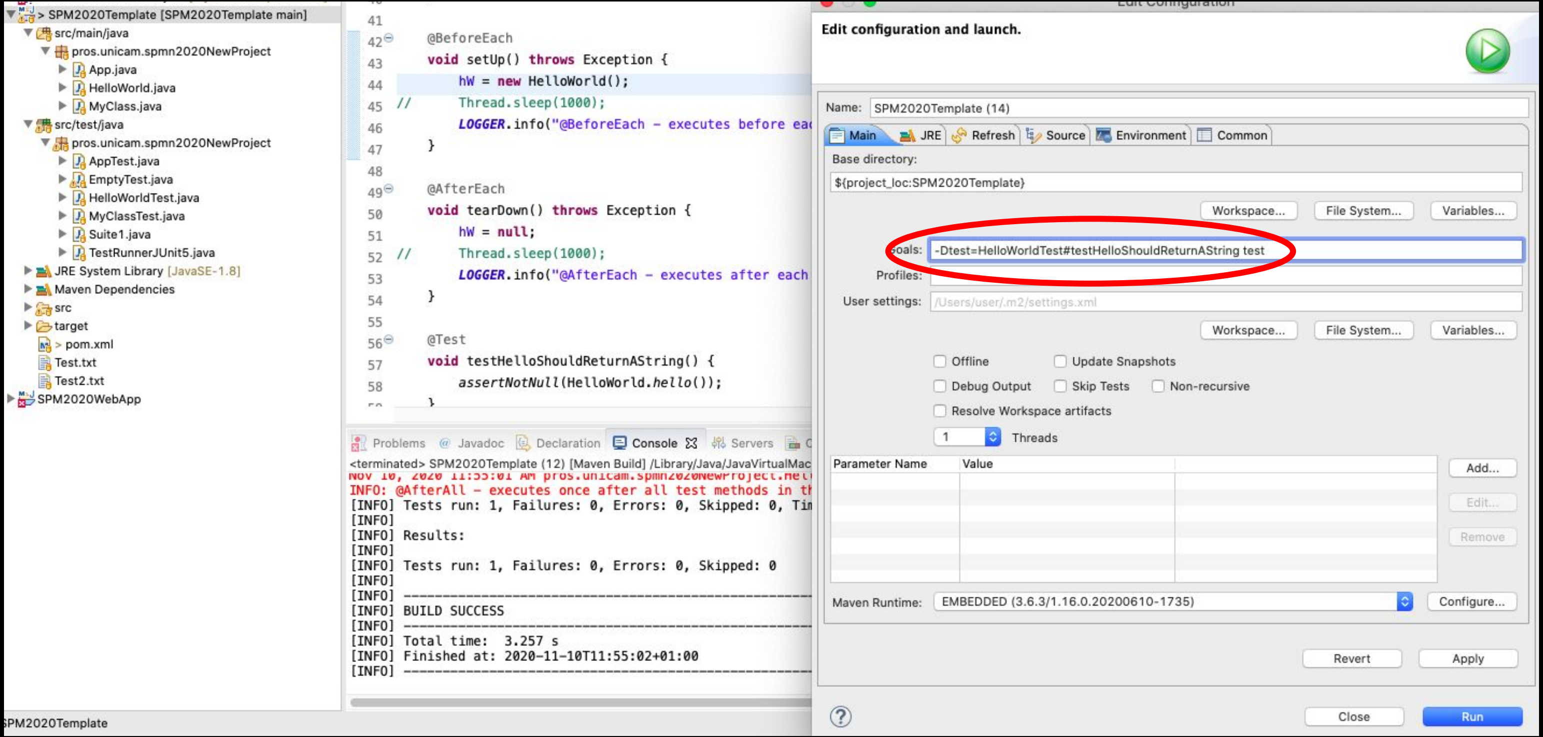
1 Threads

Parameter Name	Value

Maven Runtime: EMBEDDED (3.6.3/1.16.0.20200610-1735)

Manual Testing with Maven

- Run a single test method from a test class:
  -Dtest=<NameOfTheTestClass>#<NameOfTheTestMethod> test



```java

    @BeforeEach
    void setUp() throws Exception {
        hW = new HelloWorld();
//      Thread.sleep(1000);
        LOGGER.info("@BeforeEach - executes before eac
    }

    @AfterEach
    void tearDown() throws Exception {
        hW = null;
//      Thread.sleep(1000);
        LOGGER.info("@AfterEach - executes after each
    }

    @Test
    void testHelloShouldReturnAString() {
        assertNotNull(HelloWorld.hello());
    }
```

```
<terminated> SPM2020Template (12) [Maven Build] /Library/Java/JavaVirtualMac
Nov 10, 2020 11:55:01 AM pros.unicam.spmn2020NewProject.Hell
INFO: @AfterAll - executes once after all test methods in th
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Tim
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] ------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------
[INFO] Total time:  3.257 s
[INFO] Finished at: 2020-11-10T11:55:02+01:00
[INFO] ------------------------------------------------------
```

Edit Configuration

Edit configuration and launch.

Name: SPM2020Template (14)

Main | JRE | Refresh | Source | Environment | Common

Base directory: ${project_loc:SPM2020Template}

Workspace... File System... Variables...

Goals: -Dtest=HelloWorldTest#testHelloShouldReturnAString test

Profiles:

User settings: /Users/user/.m2/settings.xml

Workspace... File System... Variables...

☐ Offline ☐ Update Snapshots

☐ Debug Output ☐ Skip Tests ☐ Non-recursive

☐ Resolve Workspace artifacts

1 Threads

Parameter Name	Value

Add... Edit... Remove

Maven Runtime: EMBEDDED (3.6.3/1.16.0.20200610-1735) Configure...

Revert Apply

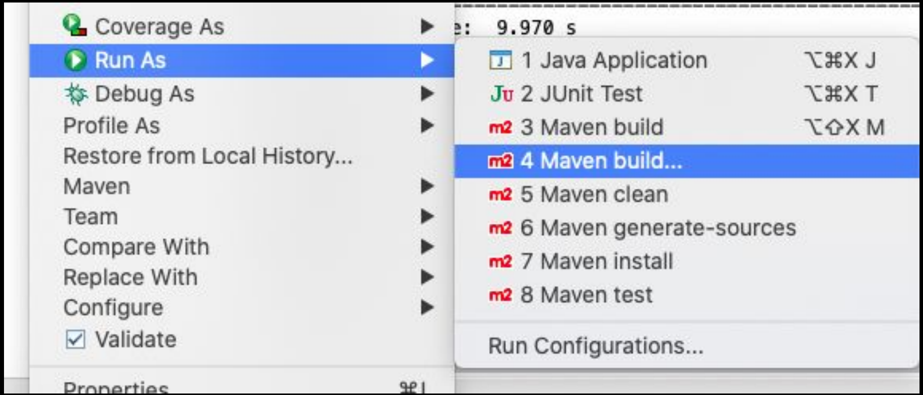
Close Run

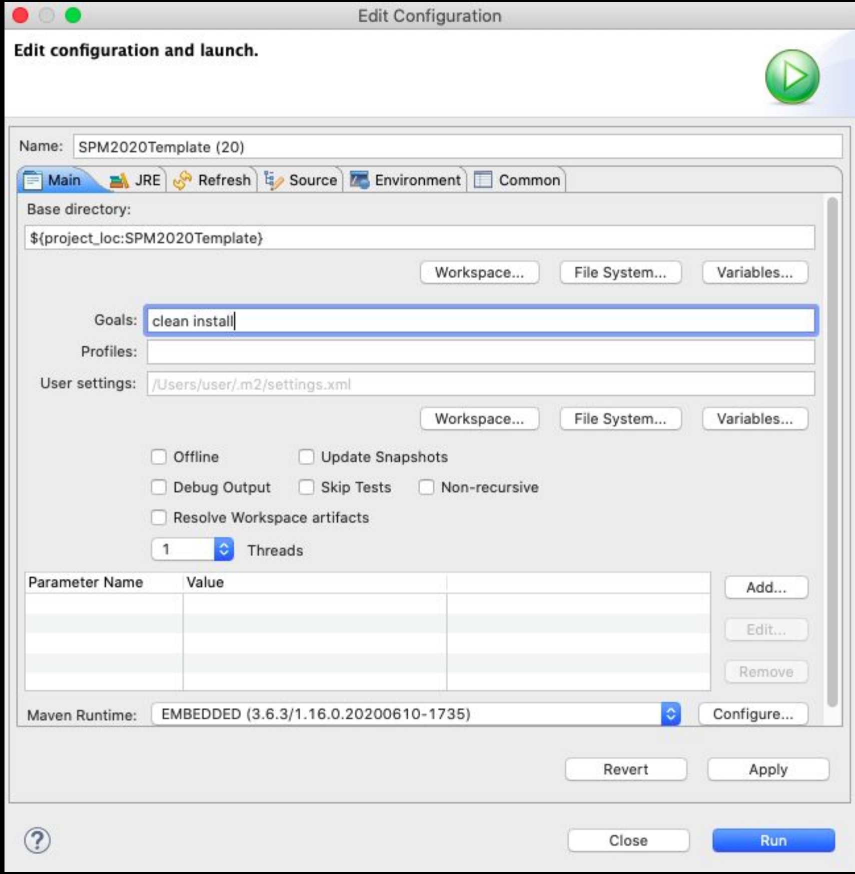
Automated testing

To automatically verify main functionality, ensure new version does not cause new defects, provide regression testing and help the teams to run a large number of tests in a short period of time.

Companies having great number of projects are looking for specialists in the field of automated testing.

Automated Testing with Maven





Name: SPM2020Template (20)

Base directory: ${project_loc:SPM2020Template}

Goals: clean install

Profiles:

User settings: /Users/user/.m2/settings.xml

□ Offline □ Update Snapshots

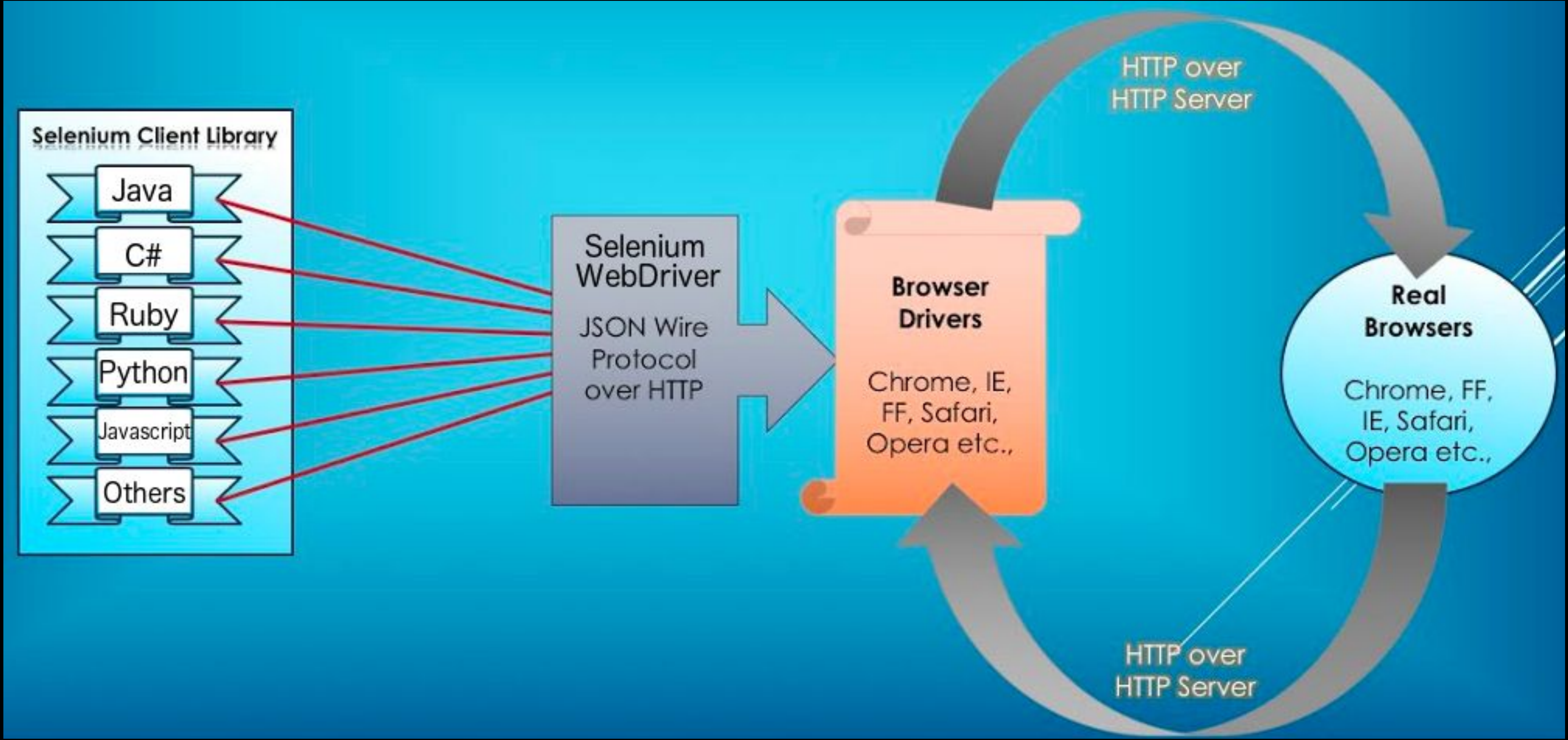
□ Debug Output □ Skip Tests □ Non-recursive

□ Resolve Workspace artifacts

1 Threads

Maven Runtime: EMBEDDED (3.6.3/1.16.0.20200610-1735)

Selenium Architecture



Selenium WebDriver

- A Selenium Web driver must be created

- For using Chrome:

```
System.setProperty("webdriver.chrome.driver",projectPath+"/drivers/chromedriver");*
WebDriver driver = new ChromeDriver();
```

- Interaction with the Chrome instance will be made in the code on the driver.

*Note: you need to specify, before instantiating the BrowserDriver, the path to the actual driver that you downloaded following instructions from the selenium website https://www.seleniumhq.org/download/.

Selenium WebDriver

- Navigation using a Selenium WebDriver is very simple, given a defined URL. It can be done in two ways, driver.get(…) or driver.navigate().to(…)

    - *driver.get("https://www.google.com/");*
    - *driver.navigate().to("https://www.google.com/");*

- *The driver.get(…) and driver.navigate().to(…) do exactly the same thing. driver.navigate() supports also driver.navigate().forward() and driver.navigate().backward()*

Finding Web Elements

- An example:

  - Assuming that we have the following web page:

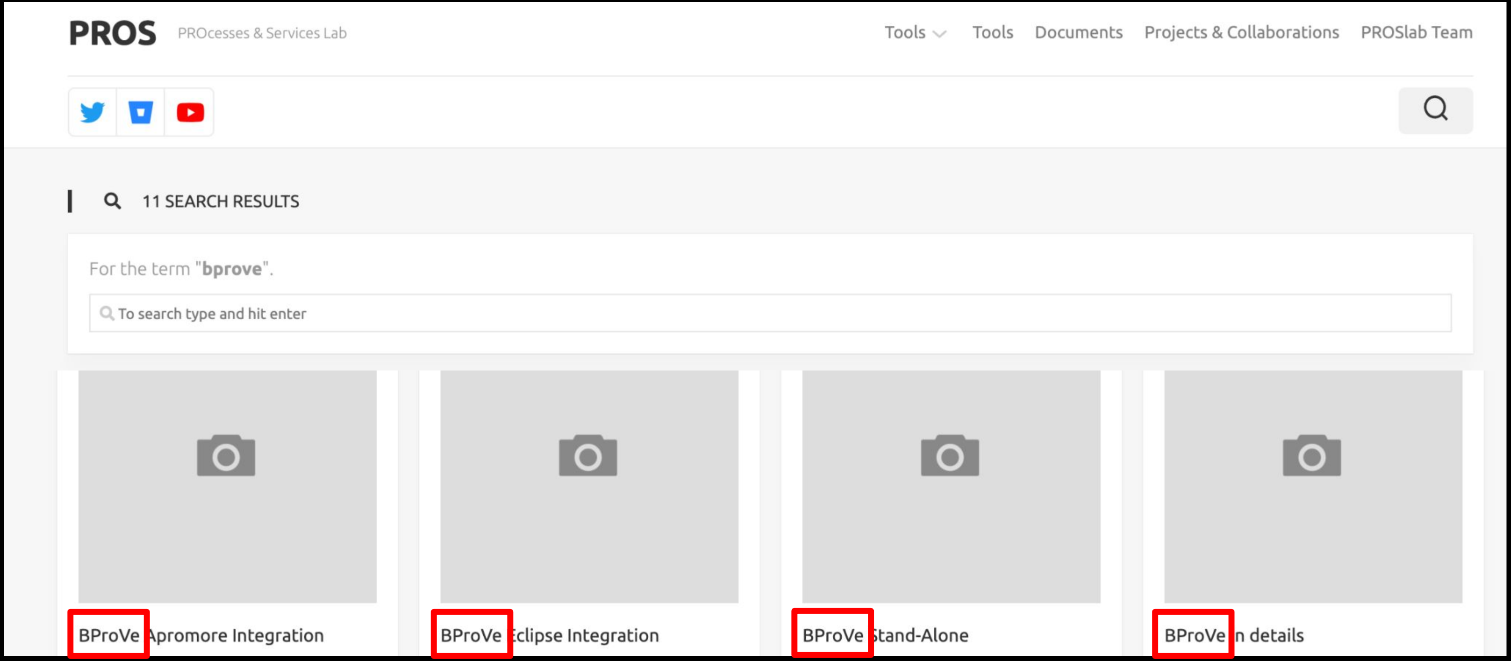
    ```
    <html>
        <body>
            <button id= “my_button”> Click Me</button>
        </body>
    </html>
    ```

  - The following lines of code will be used for clicking the button:

    ```
    WebElement button = driver.findElement(By.id(" my_button "));
    button.click();
    ```

Selenium Example

Complete the test checkProsSiteSearch to test if the search functionality on the pros.unicam.it website returns what expected. We expect to search for “bprove” and to have only results that include in the title the “bprove” term.



11 SEARCH RESULTS

For the term "**bprove**".

BProVe Apromore Integration

BProVe Eclipse Integration

BProVe Stand-Alone

BProVe in details

Lecture 10 (Review Sprint 0)

Today:

- 11 groups

The objectives:

- Checking the Team Status
- Checking the User Stories definition
- Asking and Answering Questions
- **Defining a Sprint Backlog**

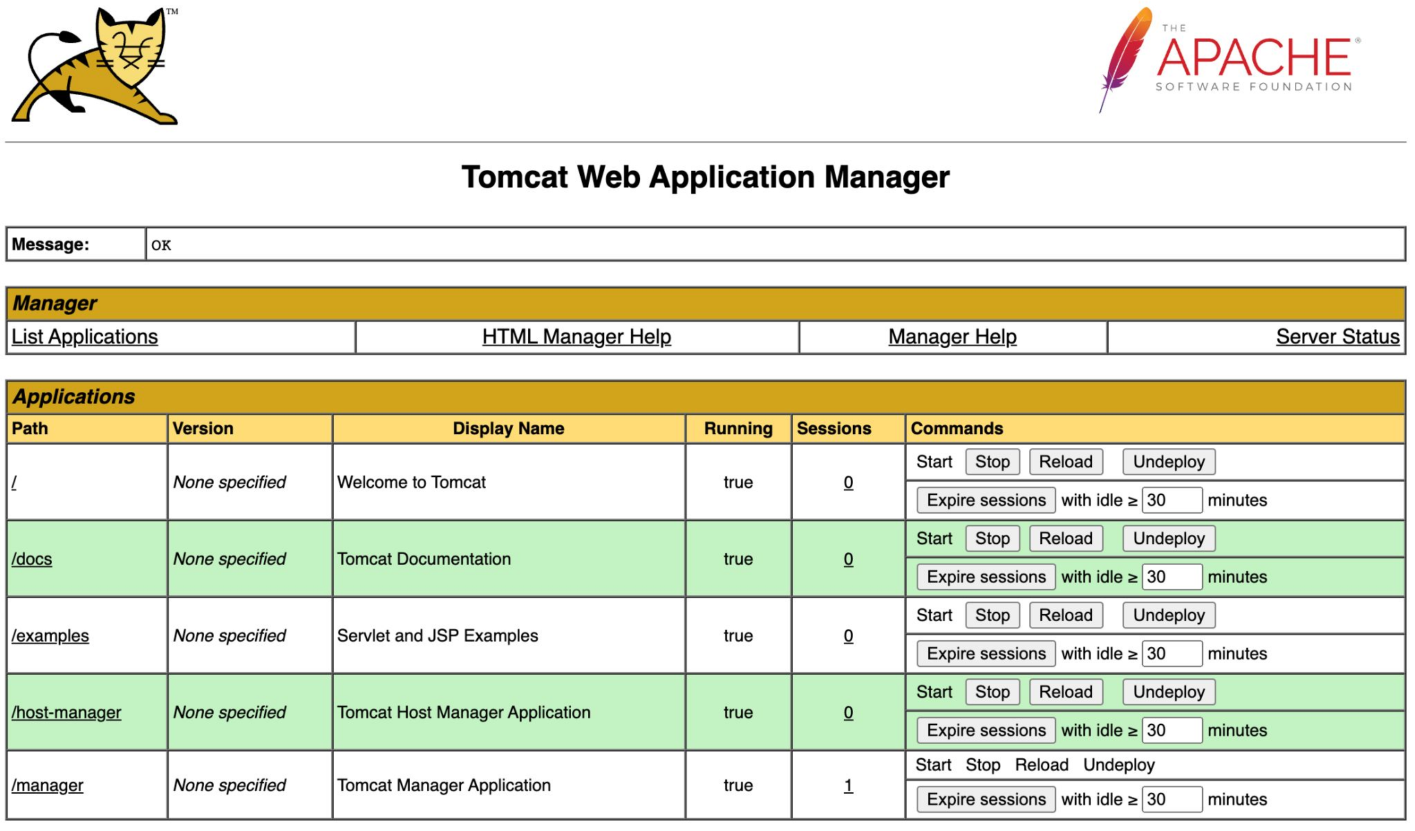
Lecture 11

The Apache Tomcat® software is an open source implementation of the Java Servlet, JavaServer Pages, Java Expression Language and Java WebSocket technologies. It is a HTTP web server environment in which Java code can run.

http://tomcat.apache.org/

Download Tomcat
https://tomcat.apache.org/download-90.cgi

Tomcat Manager



Tomcat Web Application Manager

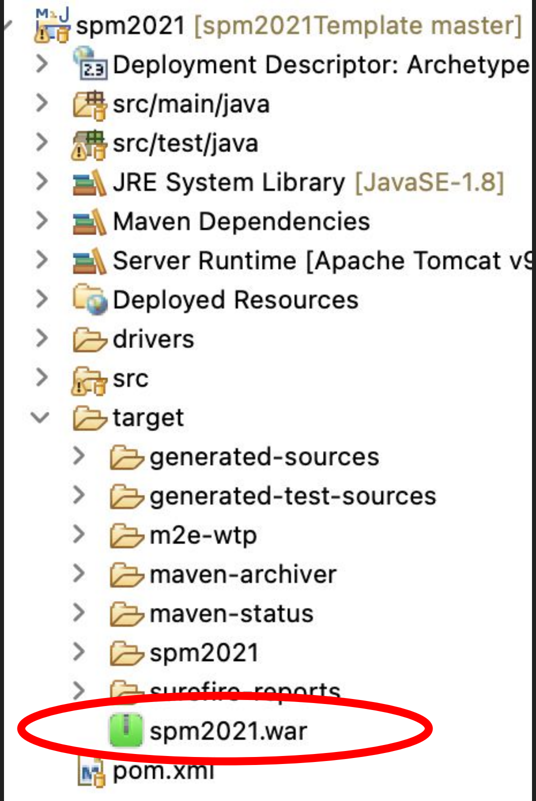
Message:	OK

Manager			
List Applications	HTML Manager Help	Manager Help	Server Status

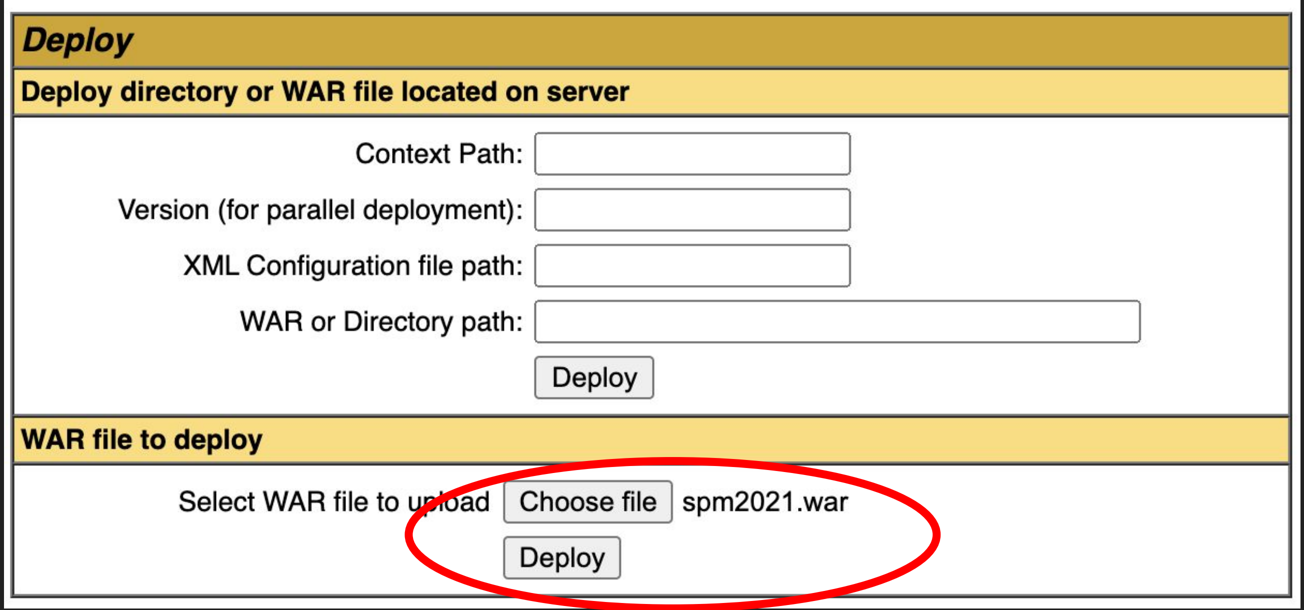
Applications

Path	Version	Display Name	Running	Sessions	Commands
/	*None specified*	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	*None specified*	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	*None specified*	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	*None specified*	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	*None specified*	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Tomcat Manager



- spm2021 [spm2021Template master]
  - Deployment Descriptor: Archetype
  - src/main/java
  - src/test/java
  - JRE System Library [JavaSE-1.8]
  - Maven Dependencies
  - Server Runtime [Apache Tomcat v9
  - Deployed Resources
  - drivers
  - src
  - target
    - generated-sources
    - generated-test-sources
    - m2e-wtp
    - maven-archiver
    - maven-status
    - spm2021
    - surefire-reports
    - spm2021.war
  - pom.xml



Deploy

Deploy directory or WAR file located on server

Context Path:

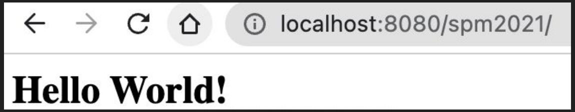
Version (for parallel deployment):

XML Configuration file path:

WAR or Directory path:

WAR file to deploy

Select WAR file to upload spm2021.war

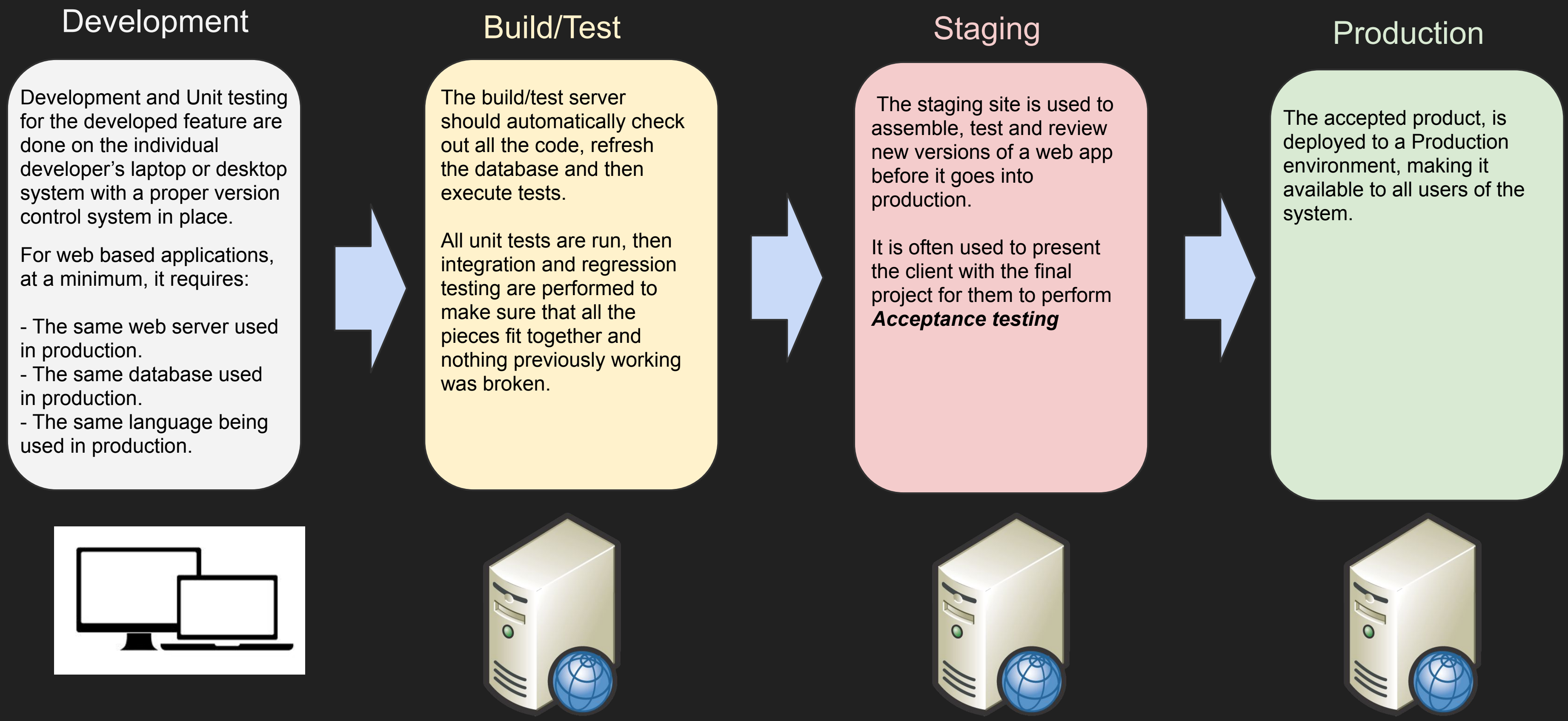


localhost:8080/spm2021/

Hello World!

Environments

NOTE: Referred also as Development, Testing, Acceptance and Production (DTAP)



Development

Development and Unit testing for the developed feature are done on the individual developer’s laptop or desktop system with a proper version control system in place.

For web based applications, at a minimum, it requires:

- The same web server used in production.
- The same database used in production.
- The same language being used in production.

Build/Test

The build/test server should automatically check out all the code, refresh the database and then execute tests.

All unit tests are run, then integration and regression testing are performed to make sure that all the pieces fit together and nothing previously working was broken.

Staging

The staging site is used to assemble, test and review new versions of a web app before it goes into production.

It is often used to present the client with the final project for them to perform ***Acceptance testing***

Production

The accepted product, is deployed to a Production environment, making it available to all users of the system.

Jenkins

Jenkins is used to build and test your product continuously, so developers can continuously integrate changes into the build.

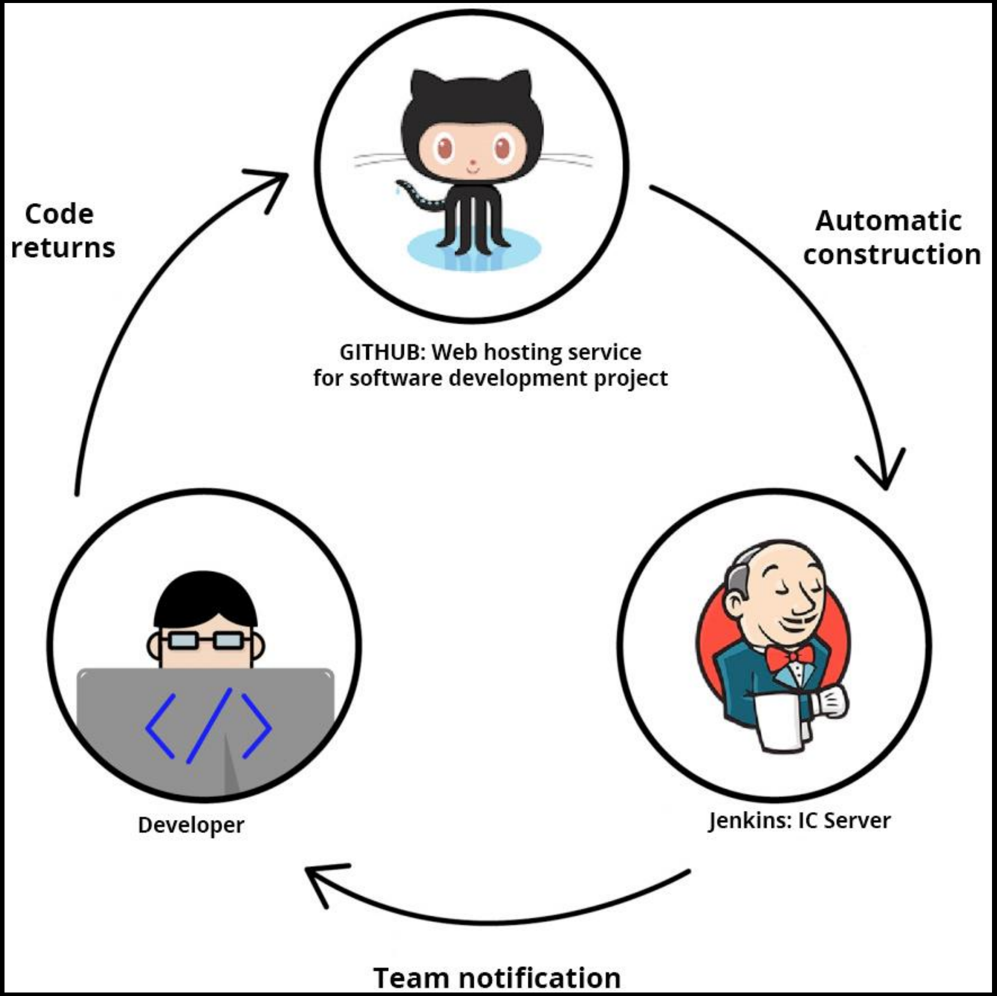
https://jenkins.io/

Jenkins Plugins

https://plugins.jenkins.io/

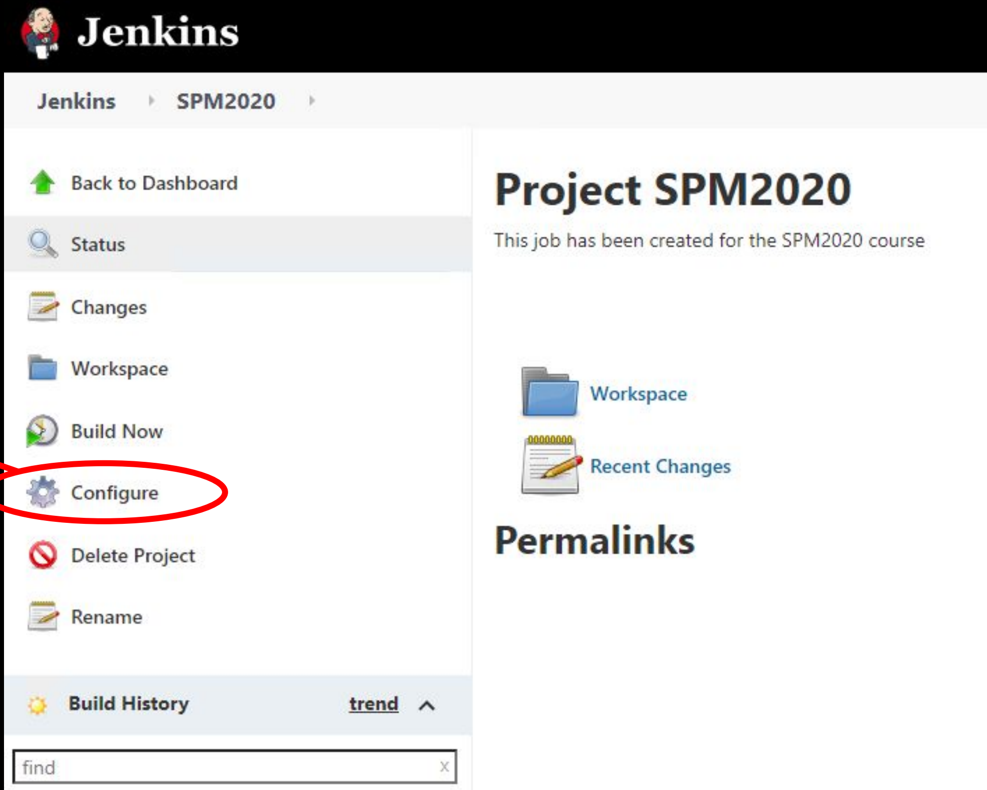
Continuous Integration with Jenkins

Jenkins triggers a build upon every commit to the source code repository, typically to a development branch.



Configure a Job

Configure for changing settings

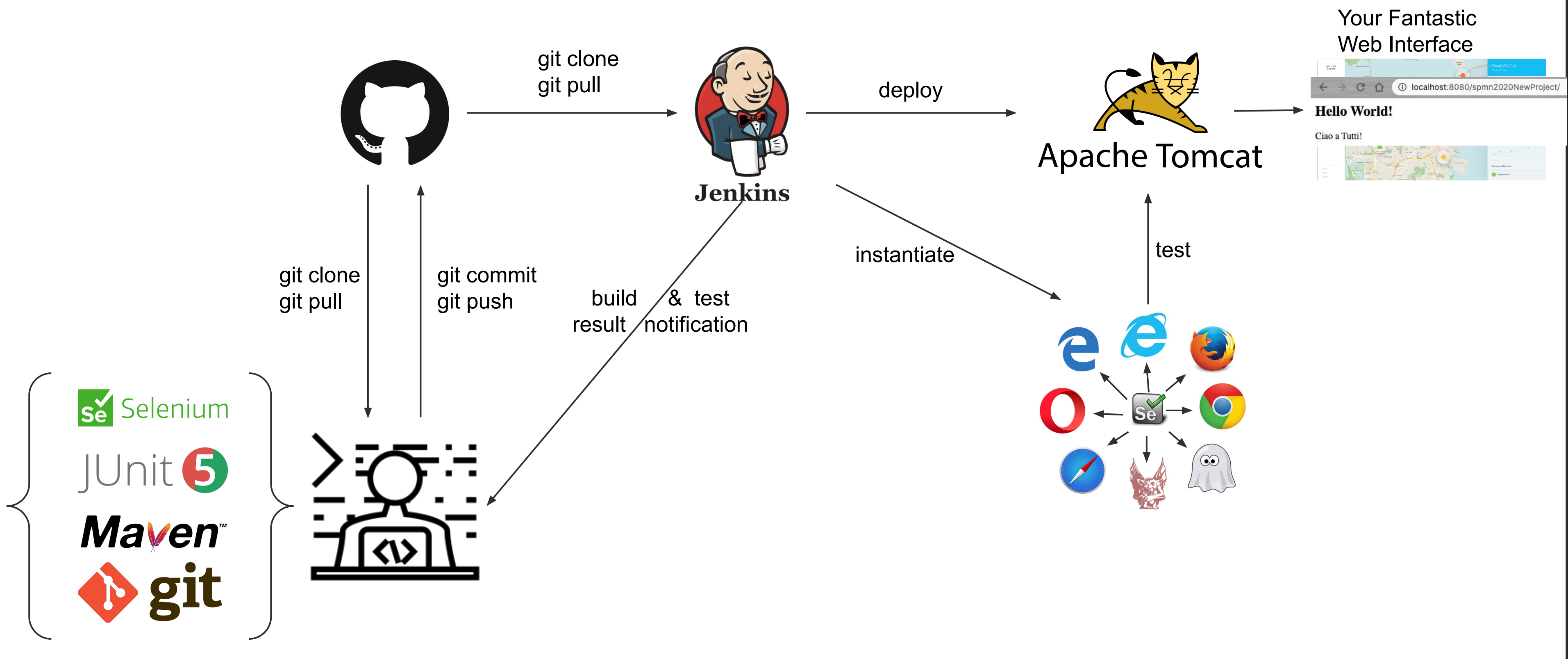


Project SPM2020

This job has been created for the SPM2020 course

Permalinks

Our DevOps Toolchain



git clone
git pull

deploy

Jenkins

Apache Tomcat

Your Fantastic
Web Interface

localhost:8080/spmn2020NewProject/

Hello World!

Ciao a Tutti!

git clone
git pull

git commit
git push

build & test
result notification

instantiate

test

Selenium

JUnit 5

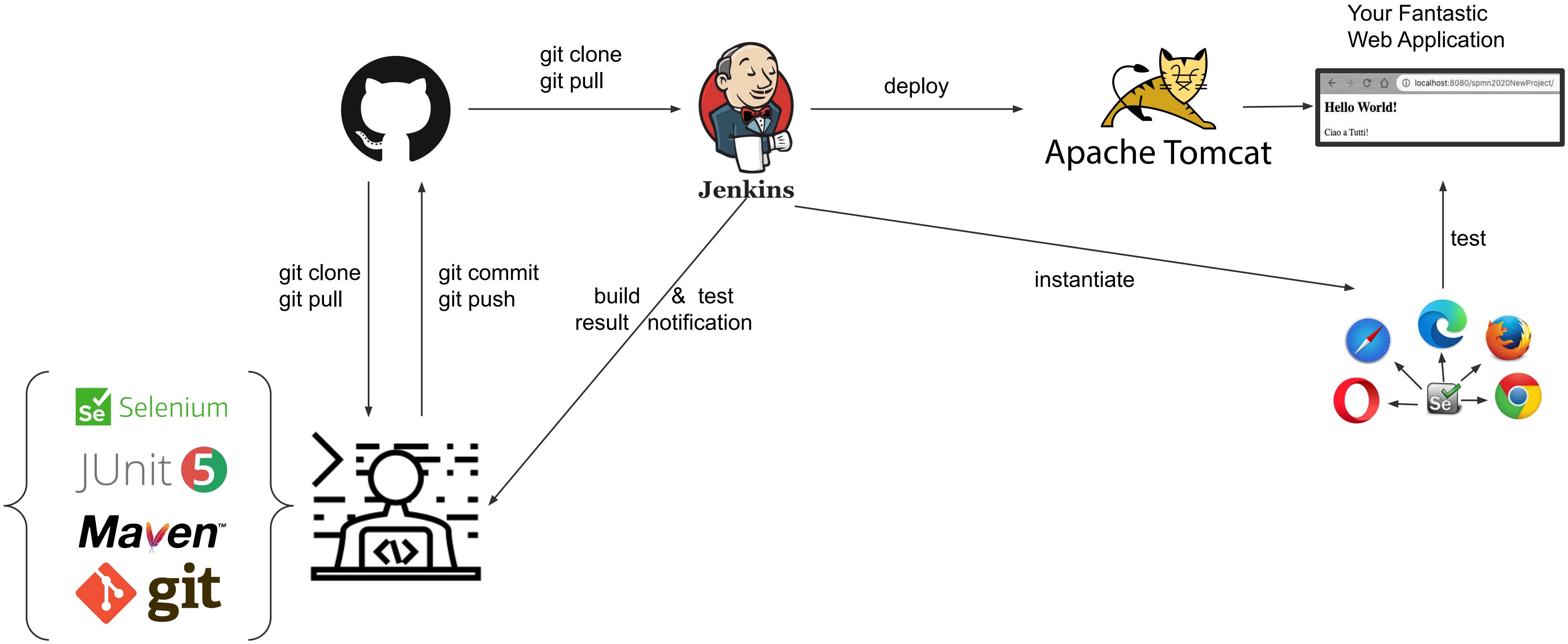
Maven™

git

Lecture 15

Acceptance Test & Headless Browsers

Our Toolchain



git clone
git pull

deploy

Your Fantastic
Web Application

localhost:8080/spmn2020NewProject/

Hello World!

Ciao a Tutti!

git clone
git pull

git commit
git push

build & test
result notification

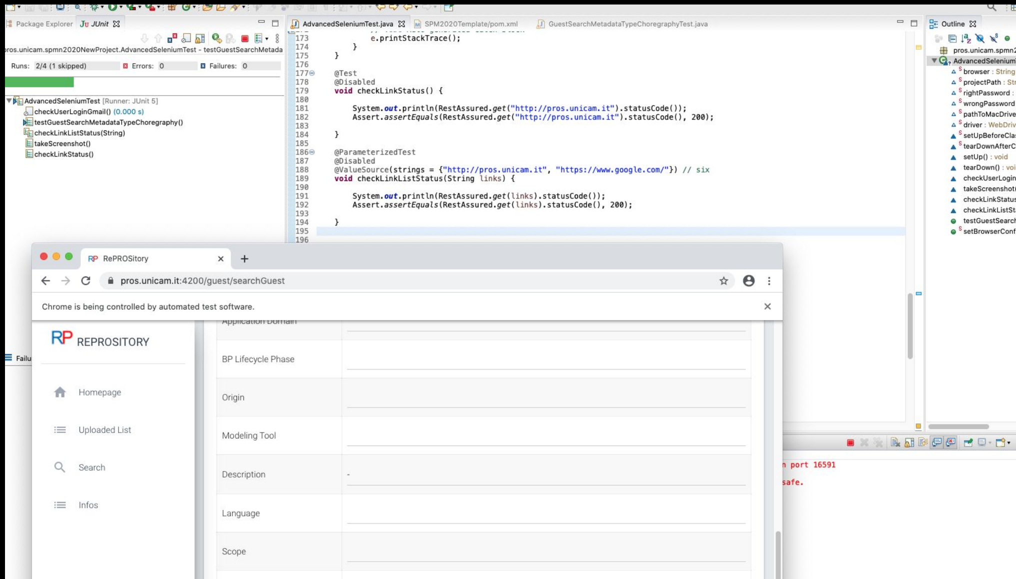
instantiate

test

Do we really need a browser…?

Or better...do we really need a graphical interface?

Every time we run a test, an instance of a browser is created and the graphical user interface of the chosen browser appears...do we really need it?



pros.unicam.spmn2020NewProject.AdvancedSeleniumTest - testGuestSearchMetada

Runs: 2/4 (1 skipped) Errors: 0 Failures: 0

- AdvancedSeleniumTest [Runner: JUnit 5]
  - checkUserLoginGmail() (0.000 s)
  - testGuestSearchMetadataTypeChoregraphy()
  - checkLinkListStatus(String)
  - takeScreenshot()
  - checkLinkStatus()

```
            e.printStackTrace();
        }
    }

    @Test
    @Disabled
    void checkLinkStatus() {

        System.out.println(RestAssured.get("http://pros.unicam.it").statusCode());
        Assert.assertEquals(RestAssured.get("http://pros.unicam.it").statusCode(), 200);

    }

    @ParameterizedTest
    @Disabled
    @ValueSource(strings = {"http://pros.unicam.it", "https://www.google.com/"}) // six
    void checkLinkListStatus(String links) {

        System.out.println(RestAssured.get(links).statusCode());
        Assert.assertEquals(RestAssured.get(links).statusCode(), 200);

    }
```

```
n port 16591
safe.
```

Chrome is being controlled by automated test software.

Application Domain

BP Lifecycle Phase

Origin

Modeling Tool

Description -

Language

Scope

Headless Browser…

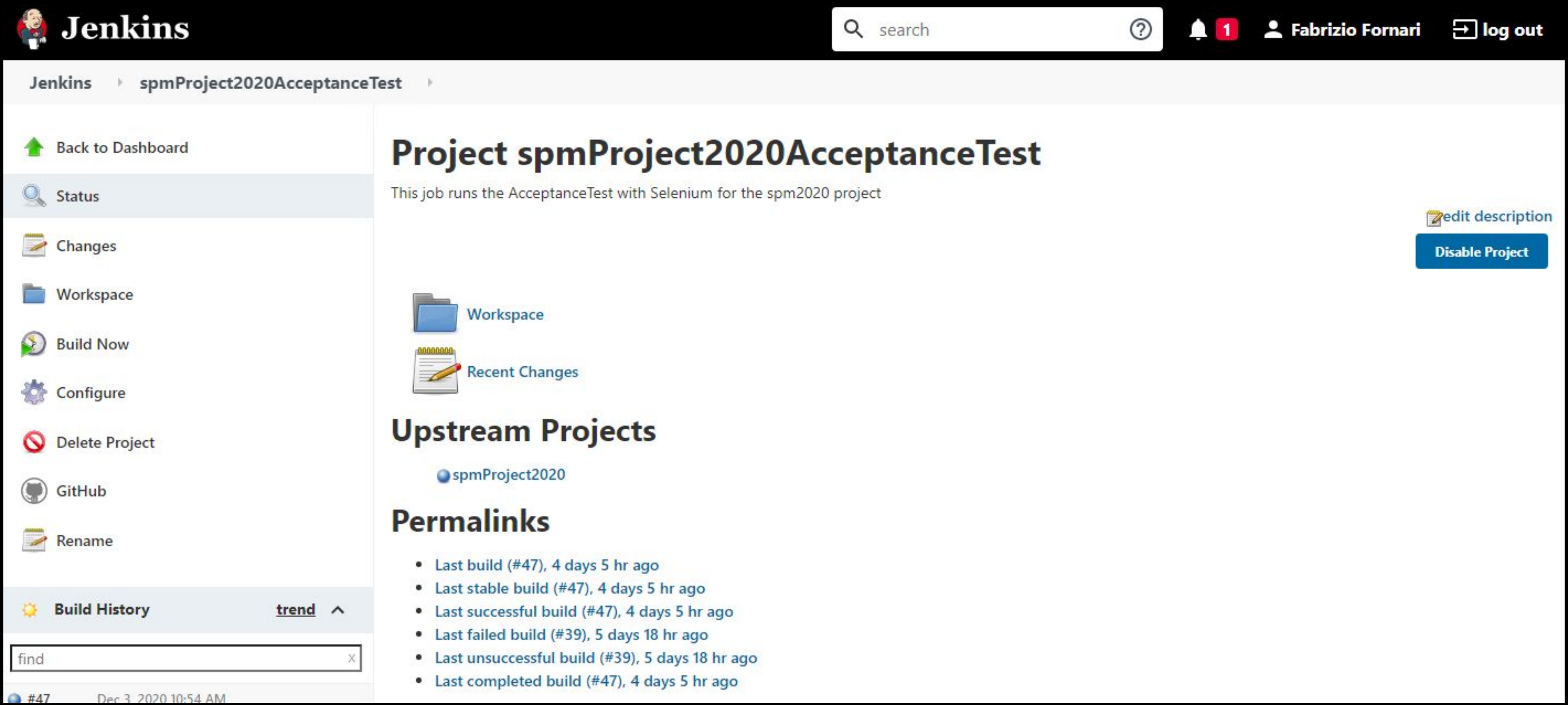
Headless Browser…

It is a browser without graphical interface

Headless browsers are commonly used for:

- Website and application testing
- JavaScript library testing
- JavaScript simulation and interactions
- Running one or more automated UI tests in the background

Create the Second Job



#47 Dec 3, 2020 10:54 AM

Project spmProject2020AcceptanceTest

This job runs the AcceptanceTest with Selenium for the spm2020 project

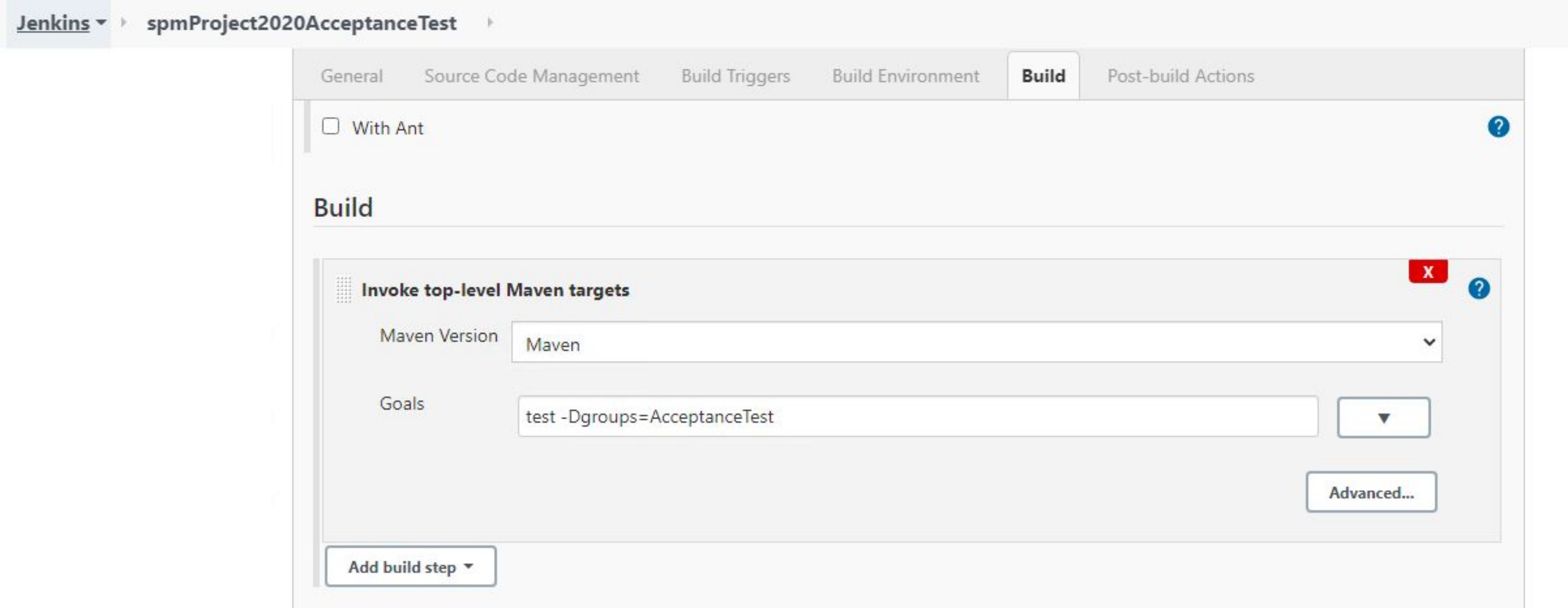
Upstream Projects

spmProject2020

Permalinks

- Last build (#47), 4 days 5 hr ago
- Last stable build (#47), 4 days 5 hr ago
- Last successful build (#47), 4 days 5 hr ago
- Last failed build (#39), 5 days 18 hr ago
- Last unsuccessful build (#39), 5 days 18 hr ago
- Last completed build (#47), 4 days 5 hr ago

Configure the Second Job



Jenkins ▾ › spmProject2020AcceptanceTest ›

□ With Ant

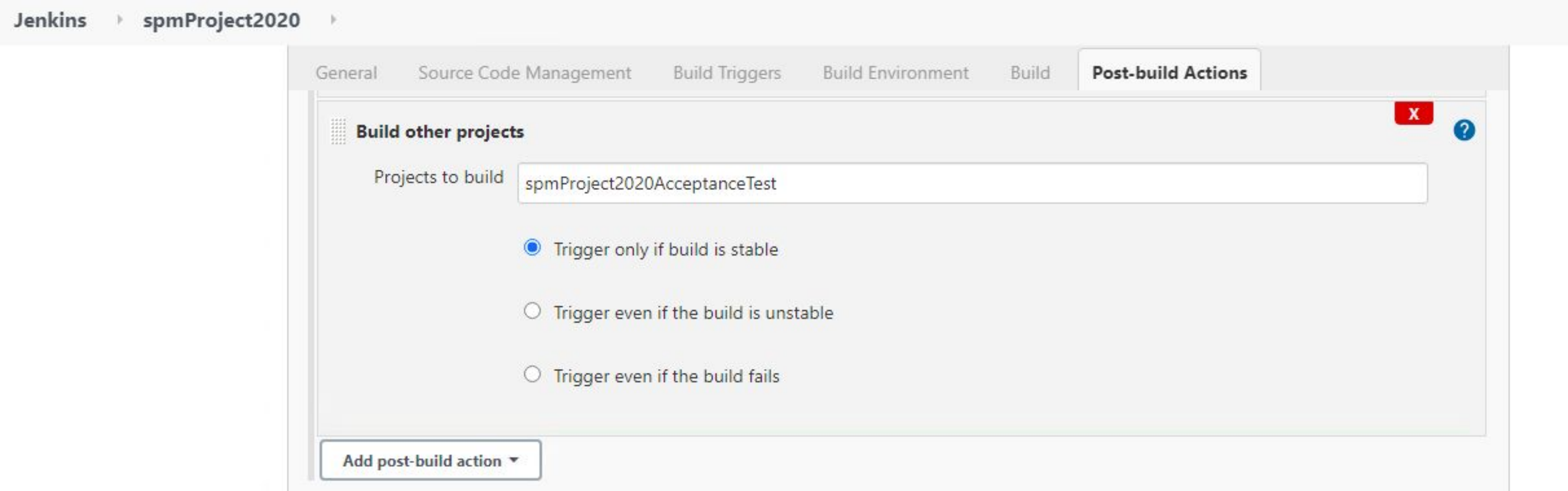
Build

Invoke top-level Maven targets

Maven Version: Maven

Goals: test -Dgroups=AcceptanceTest

Modify the First Job



Build other projects

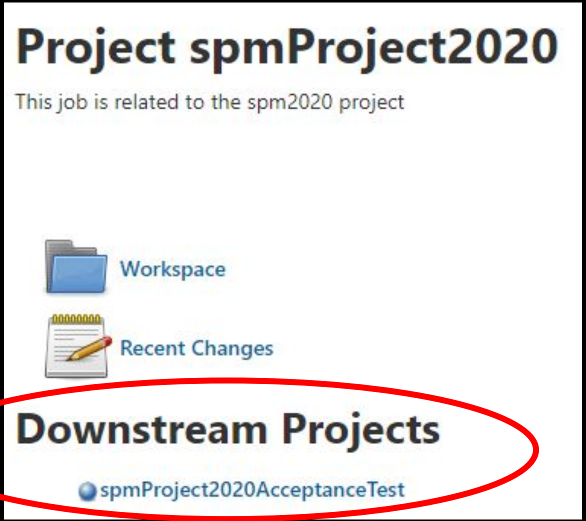
Projects to build: spmProject2020AcceptanceTest

● Trigger only if build is stable

○ Trigger even if the build is unstable

○ Trigger even if the build fails

Downstream/Upstream

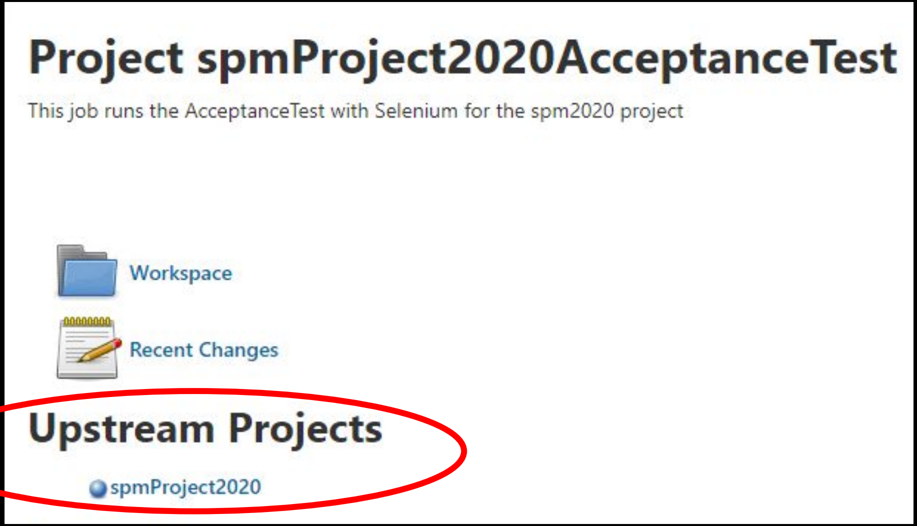


Project spmProject2020

This job is related to the spm2020 project

Downstream Projects

spmProject2020AcceptanceTest



Project spmProject2020AcceptanceTest

This job runs the AcceptanceTest with Selenium for the spm2020 project

Upstream Projects

spmProject2020

What about complex tests…?

Do we have to write them entirely from scratch?

Fortunately No!

Selenium IDE

Download it from:

https://www.seleniumhq.org/selenium-ide/

and let us see what we can do with it...

However we cannot export tests in a format that we can use for writing tests in our preferred programming language

Katalon Recorder

Katalon Automation Recorder it is an automation recorder that helps to export Selenium WebDriver code.

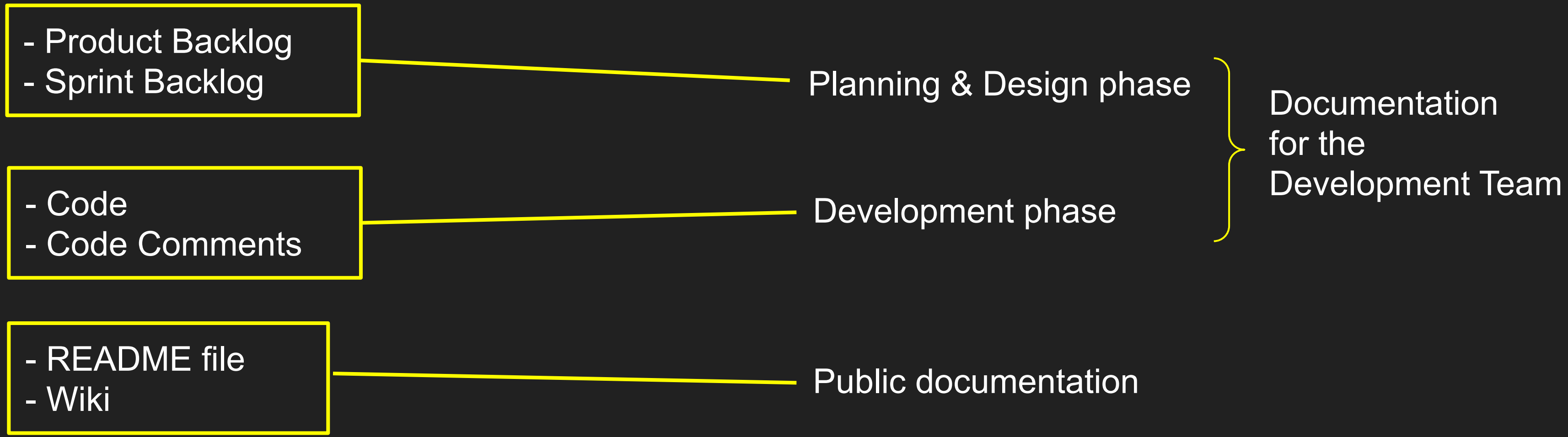
Download the extension for the browser you want to use

Explore testGuestSearchMetadataTypeChoregraphy method

Katalon

https://www.katalon.com/

Lecture 17



- Product Backlog
- Sprint Backlog

Planning & Design phase

- Code
- Code Comments

Development phase

Documentation for the Development Team

- README file
- Wiki

Public documentation

README

You can add a README file to a repository to communicate important information about your project. A README, along with a repository license, contribution guidelines, and a code of conduct, communicates expectations for your project and helps you manage contributions

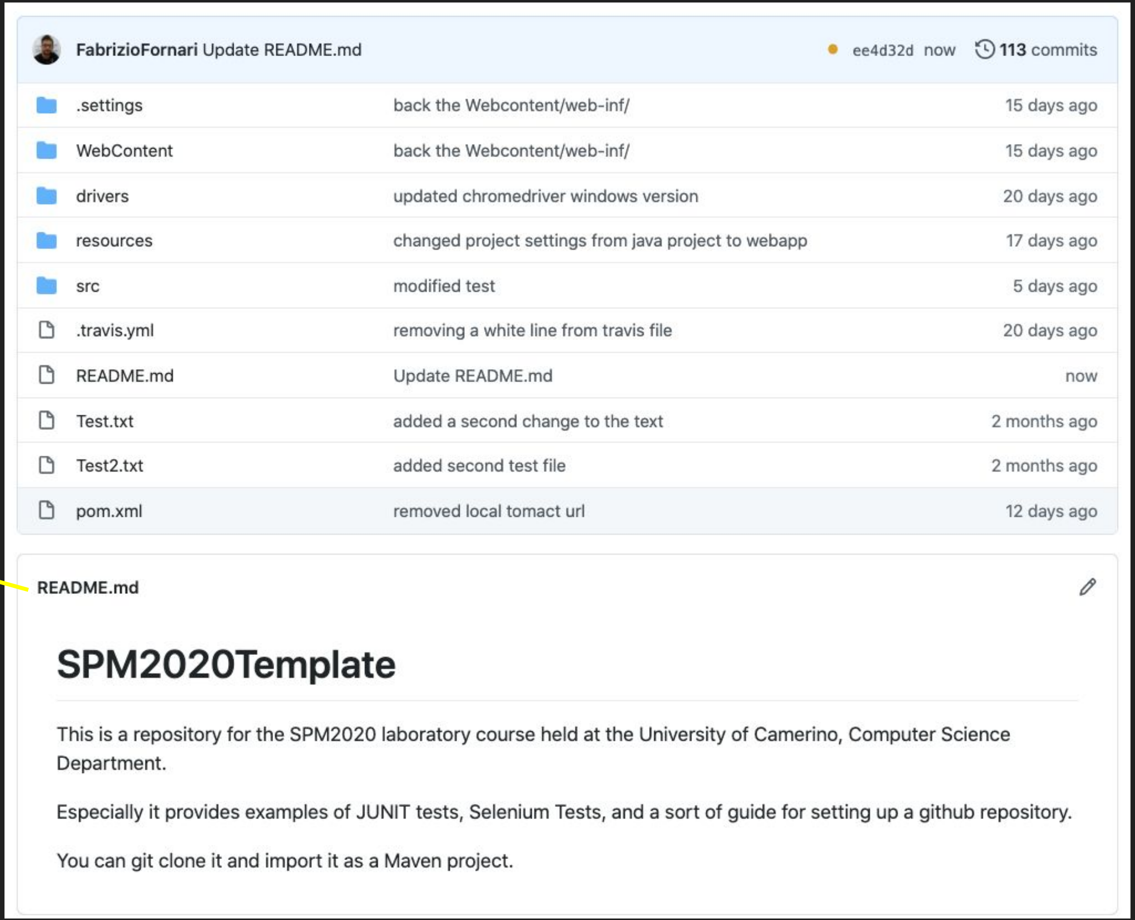
A README is often the first item a visitor will see when visiting your repository. README files typically include information on:

- What the project does
- Why the project is useful
- How users can get started with the project
- Where users can get help with your project
- Who maintains and contributes to the project

If you put your README file in your repository's root, `docs`, or hidden `.github` directory, GitHub will recognize and automatically surface your README to repository visitors.

README

README file



FabrizioFornari Update README.md ● ee4d32d now 113 commits

.settings	back the Webcontent/web-inf/	15 days ago
WebContent	back the Webcontent/web-inf/	15 days ago
drivers	updated chromedriver windows version	20 days ago
resources	changed project settings from java project to webapp	17 days ago
src	modified test	5 days ago
.travis.yml	removing a white line from travis file	20 days ago
README.md	Update README.md	now
Test.txt	added a second change to the text	2 months ago
Test2.txt	added second test file	2 months ago
pom.xml	removed local tomact url	12 days ago

README.md

SPM2020Template

This is a repository for the SPM2020 laboratory course held at the University of Camerino, Computer Science Department.

Especially it provides examples of JUNIT tests, Selenium Tests, and a sort of guide for setting up a github repository.

You can git clone it and import it as a Maven project.

Github - Wiki

Every GitHub repository comes equipped with a section for hosting documentation, called a **wiki**. We can use our repository's wiki to share long-form content about our project, such as how to use it, how we designed it, or its core principles. We can use a wiki to provide additional documentation.

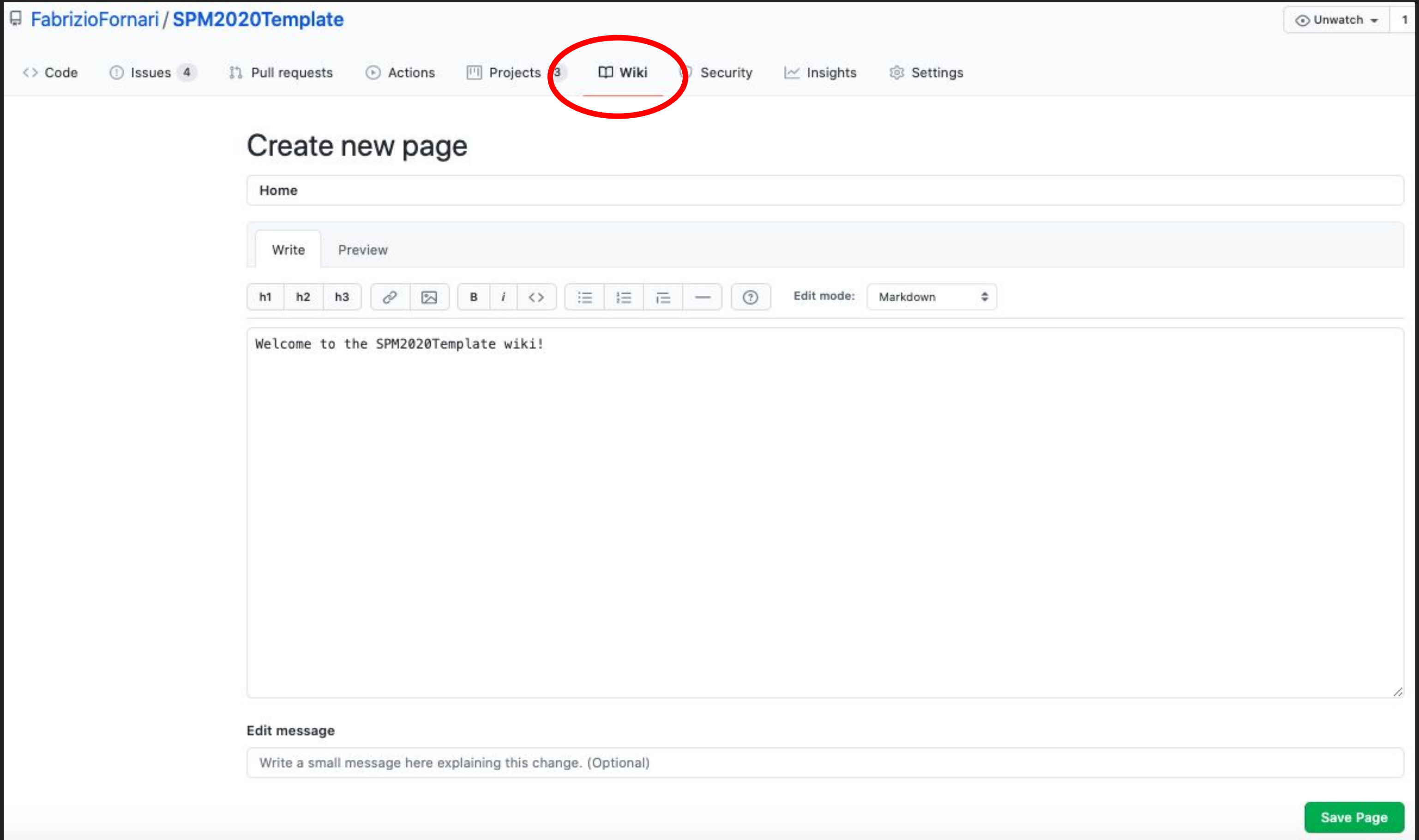
If you create a wiki in a public repository, the wiki is available to the public. If you create a wiki in an internal or private repository, people with access to the repository can also access the wiki.

You can edit wikis directly on GitHub, or you can edit wiki files locally. By default, only people with write access to your repository can make changes to wikis, although you can allow everyone on GitHub to contribute to a wiki in a public repository.

Cloning wikis to your computer

```
$ git clone https://github.com/YOUR_USERNAME/YOUR_REPOSITORY.wiki.git
# Clones the wiki locally
```

Github - Wiki



Create new page

Home

Edit mode: Markdown

```
Welcome to the SPM2020Template wiki!
```

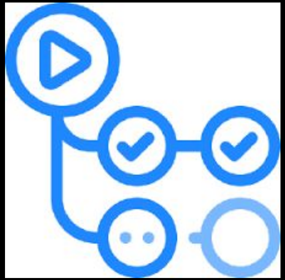
Edit message

Write a small message here explaining this change. (Optional)

Not Only Jenkins

TRAVIS CI

https://travis-ci.com/



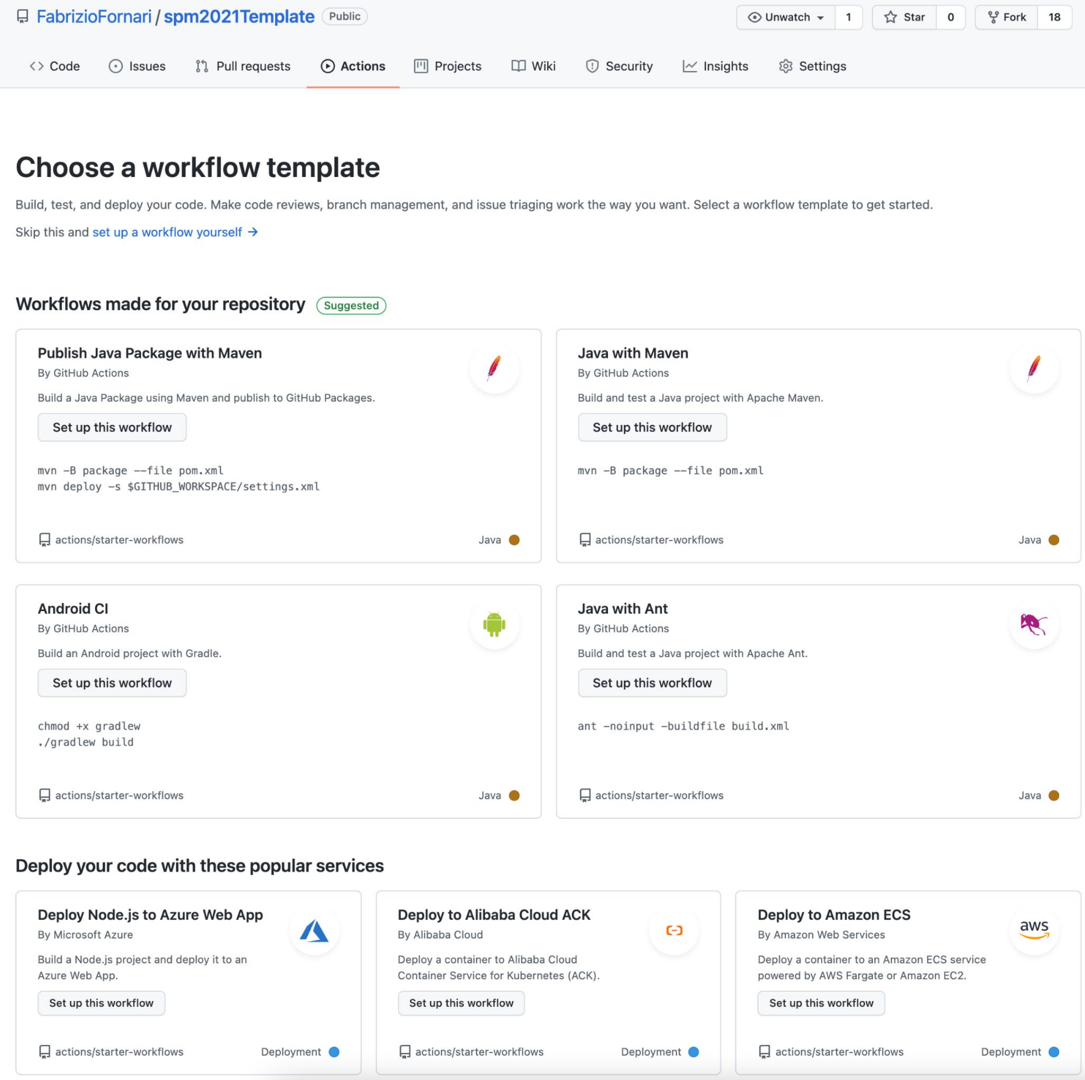
GitHub Actions

https://docs.github.com/en/actions

GitHub Actions

Get executed on GitHub Server

Jobs are execute on virtual machines hosted by GitHub.



FabrizioFornari / spm2021Template Public

Choose a workflow template

Build, test, and deploy your code. Make code reviews, branch management, and issue triaging work the way you want. Select a workflow template to get started.

Skip this and set up a workflow yourself →

Workflows made for your repository Suggested

Publish Java Package with Maven
By GitHub Actions
Build a Java Package using Maven and publish to GitHub Packages.
Set up this workflow

```
mvn -B package --file pom.xml
mvn deploy -s $GITHUB_WORKSPACE/settings.xml
```

actions/starter-workflows Java

Java with Maven
By GitHub Actions
Build and test a Java project with Apache Maven.
Set up this workflow

```
mvn -B package --file pom.xml
```

actions/starter-workflows Java

Android CI
By GitHub Actions
Build an Android project with Gradle.
Set up this workflow

```
chmod +x gradlew
./gradlew build
```

actions/starter-workflows Java

Java with Ant
By GitHub Actions
Build and test a Java project with Apache Ant.
Set up this workflow

```
ant -noinput -buildfile build.xml
```

actions/starter-workflows Java

Deploy your code with these popular services

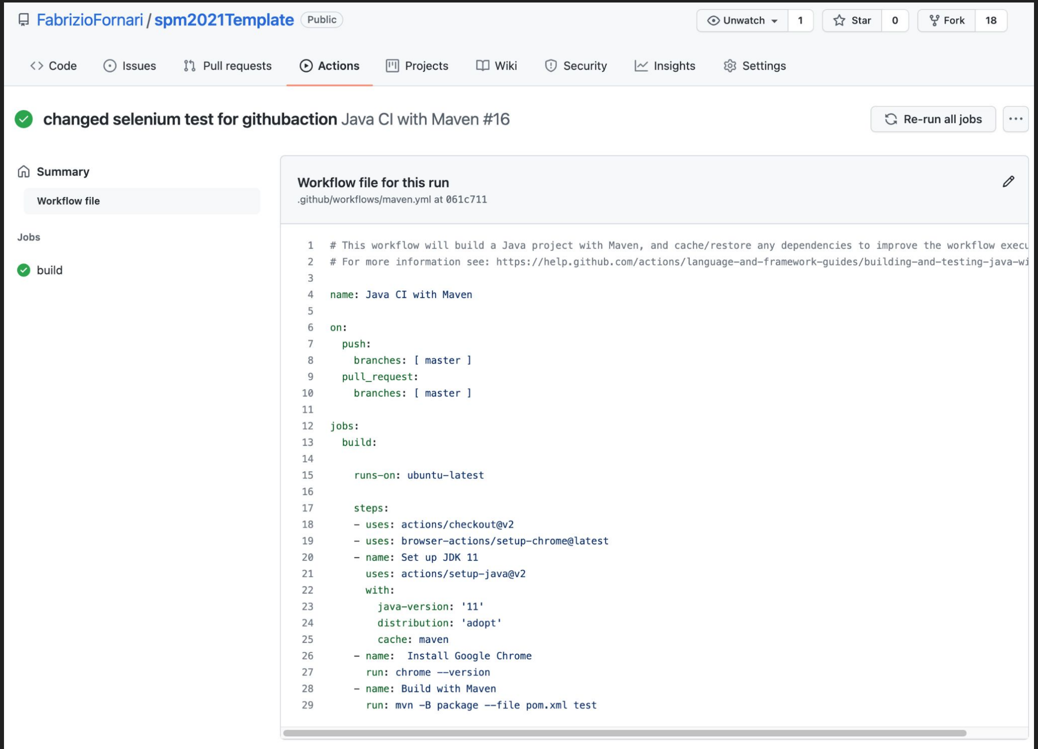
Deploy Node.js to Azure Web App
By Microsoft Azure
Build a Node.js project and deploy it to an Azure Web App.
Set up this workflow
actions/starter-workflows Deployment

Deploy to Alibaba Cloud ACK
By Alibaba Cloud
Deploy a container to Alibaba Cloud Container Service for Kubernetes (ACK).
Set up this workflow
actions/starter-workflows Deployment

Deploy to Amazon ECS
By Amazon Web Services
Deploy a container to an Amazon ECS service powered by AWS Fargate or Amazon EC2.
Set up this workflow
actions/starter-workflows Deployment

GitHub Actions

```
name: Java CI with Maven

on:
 push:
   branches: [ master ]
 pull_request:
   branches: [ master ]

jobs:
 build:

   runs-on: ubuntu-latest

   steps:
   - uses: actions/checkout@v2
   - uses: browser-actions/setup-chrome@latest
   - name: Set up JDK 11
     uses: actions/setup-java@v2
     with:
       java-version: '11'
       distribution: 'adopt'
       cache: maven
   - name: Build with Maven
     run: mvn -B package --file pom.xml test
```



changed selenium test for githubaction Java CI with Maven #16

Workflow file for this run

.github/workflows/maven.yml at 061c711

```
# This workflow will build a Java project with Maven, and cache/restore any dependencies to improve the workflow execu
# For more information see: https://help.github.com/actions/language-and-framework-guides/building-and-testing-java-wi

name: Java CI with Maven

on:
  push:
    branches: [ master ]
  pull_request:
    branches: [ master ]

jobs:
  build:

    runs-on: ubuntu-latest

    steps:
    - uses: actions/checkout@v2
    - uses: browser-actions/setup-chrome@latest
    - name: Set up JDK 11
      uses: actions/setup-java@v2
      with:
        java-version: '11'
        distribution: 'adopt'
        cache: maven
    - name:  Install Google Chrome
      run: chrome --version
    - name: Build with Maven
      run: mvn -B package --file pom.xml test
```

docker

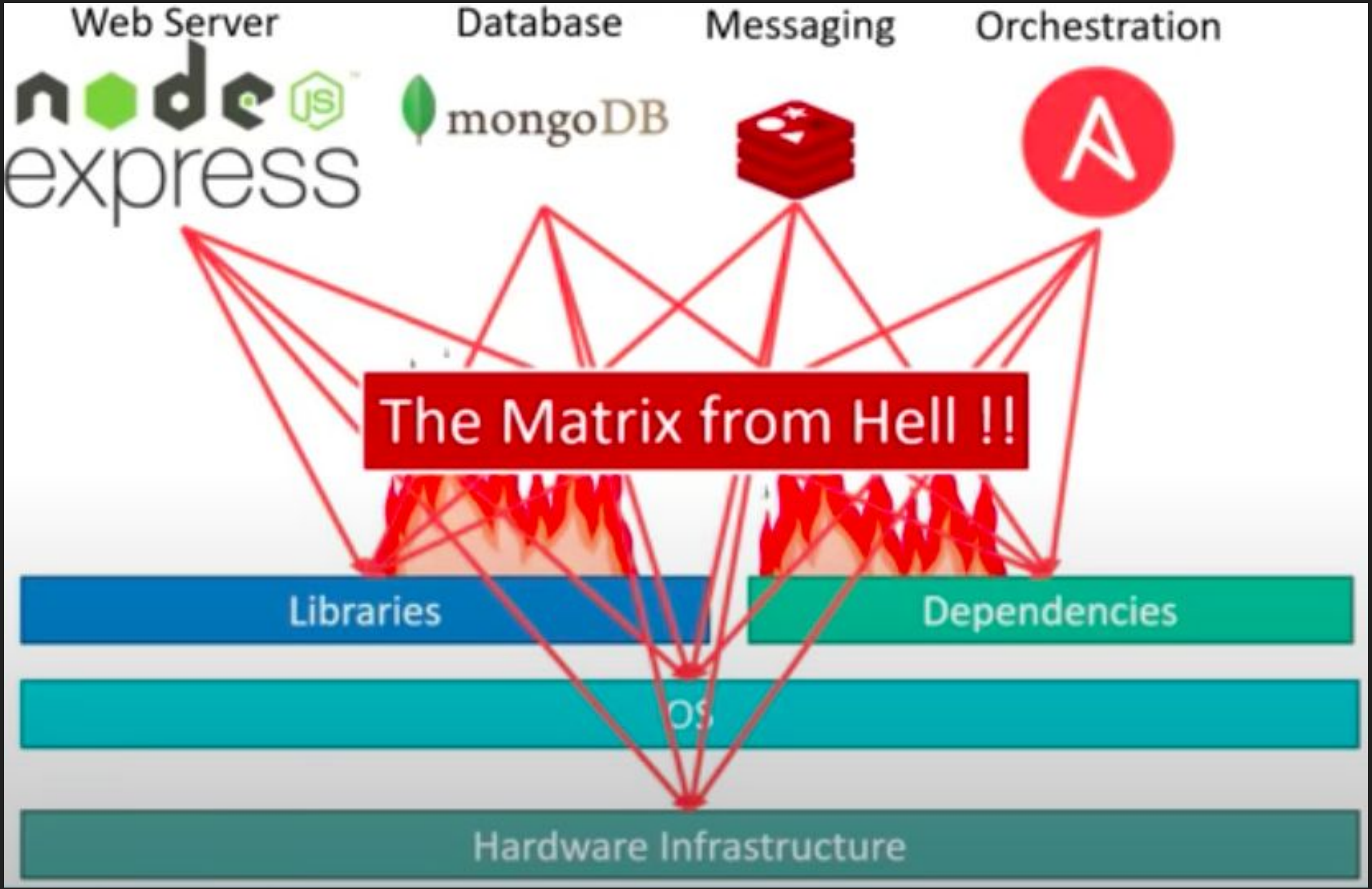
https://www.docker.com/

Docker is an open platform for developing, shipping, and running applications.
Docker enables you to separate your applications from your infrastructure so you can deliver software quickly.
With Docker, you can manage your infrastructure in the same ways you manage your applications.

Compatibility/
Dependency

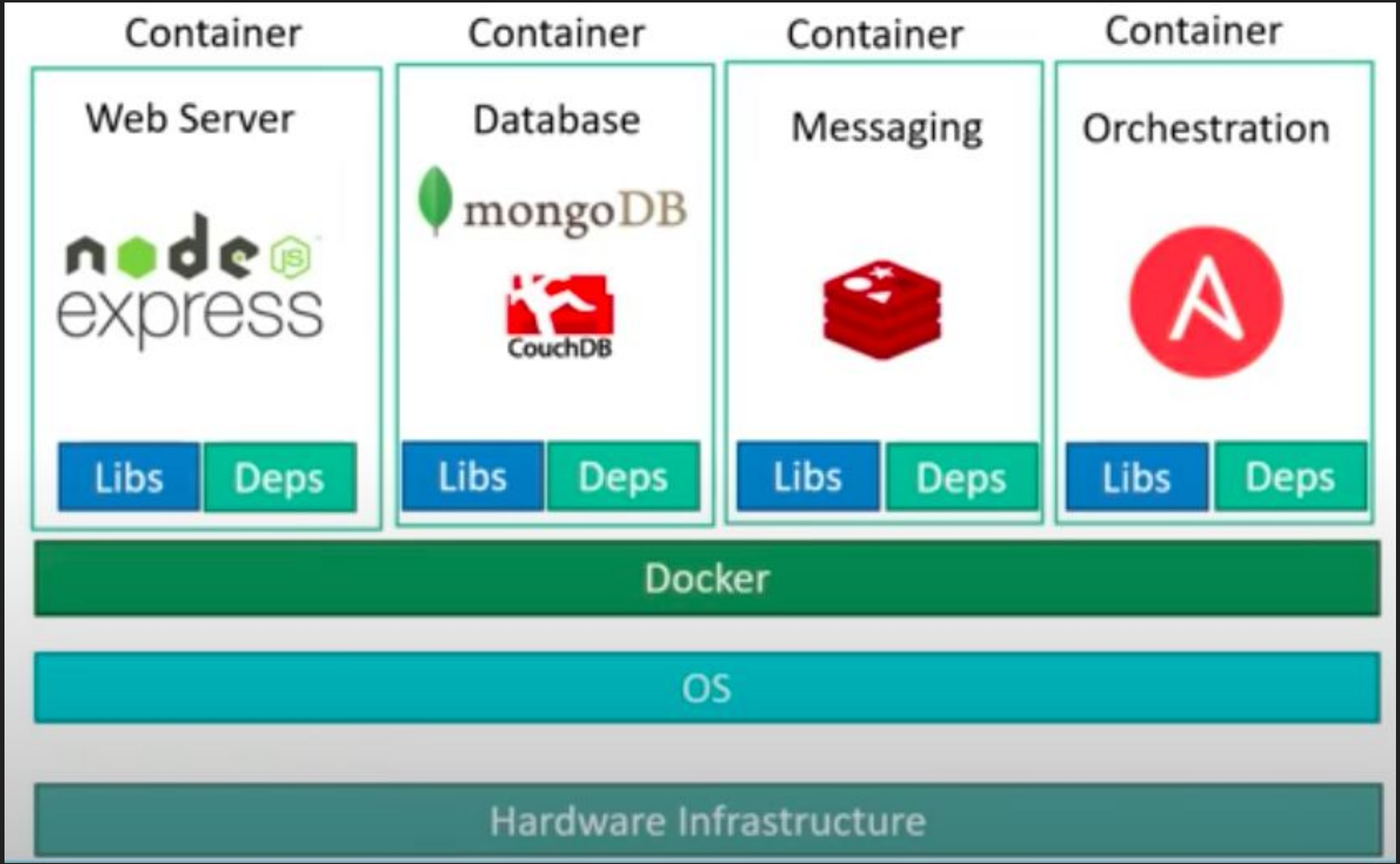
Long setup
time

Different
Dev/Test/Prod
environments

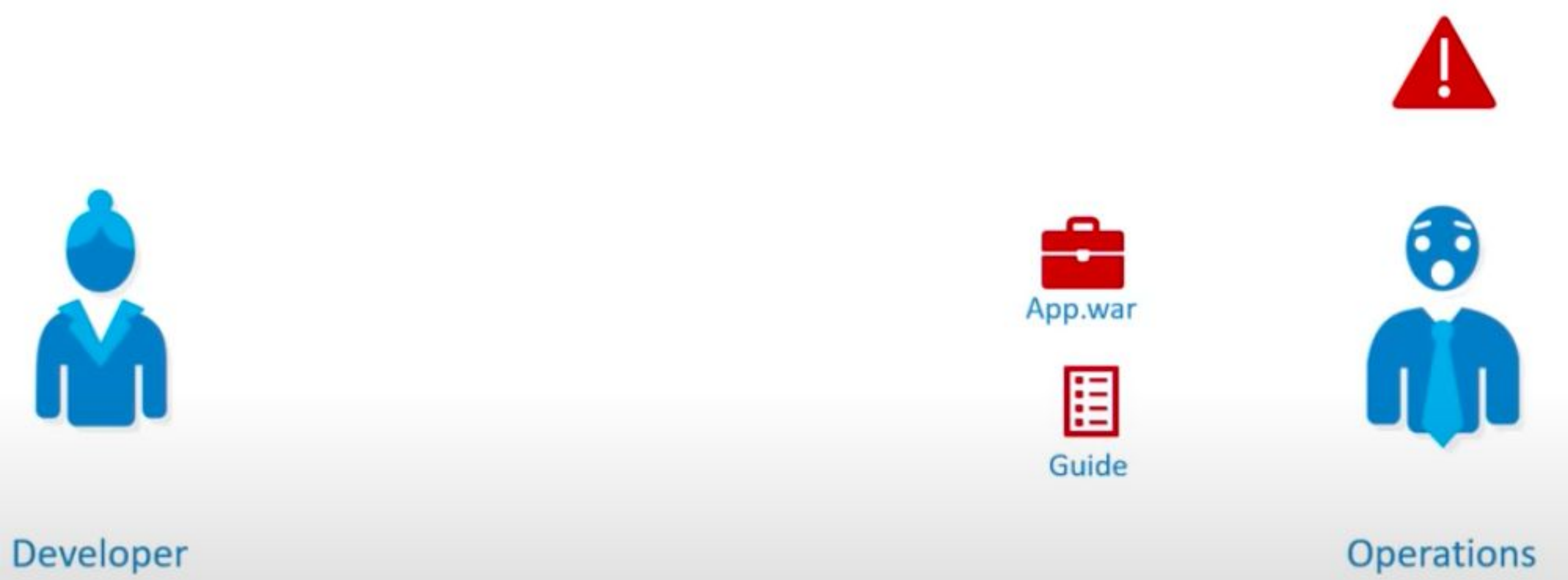


Containerized Application

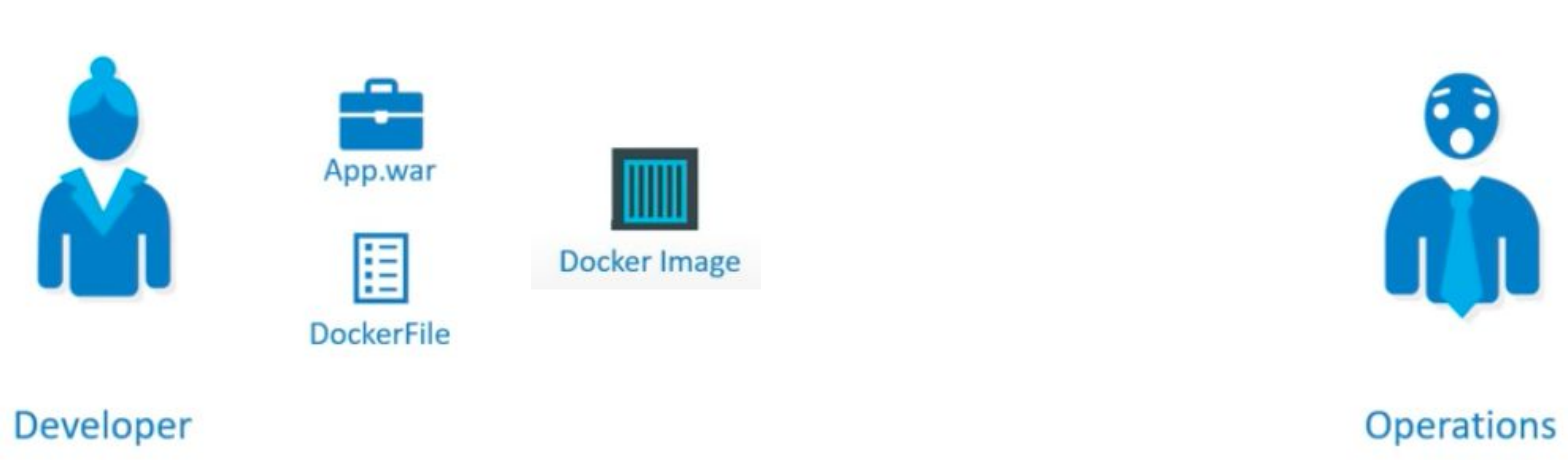
Run each service with its own dependencies in separate containers



From Application to Container



From Application to Container



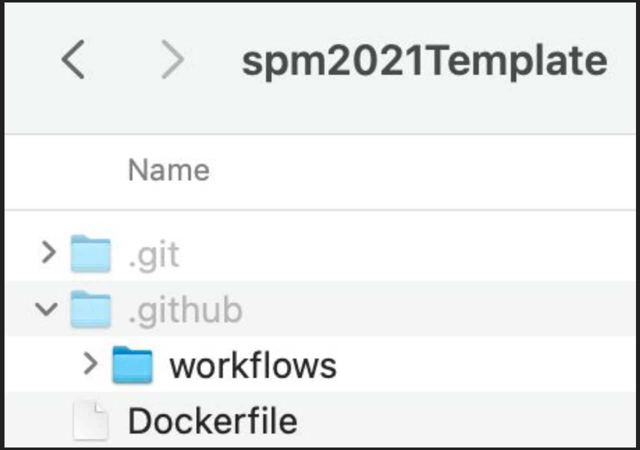
It Fixes the traditional “but it works on my machine”

From Application to Container

It Fixes the traditional “but it works on my machine”

Dockerfile

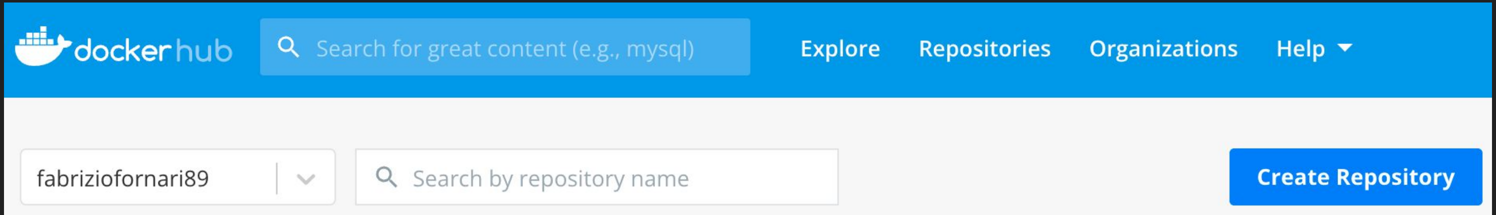
```
FROM tomcat

COPY /target/spm2021.war /usr/local/tomcat/webapps/

CMD ["catalina.sh", "run"]
```



Public Docker Images Repository

https://hub.docker.com/

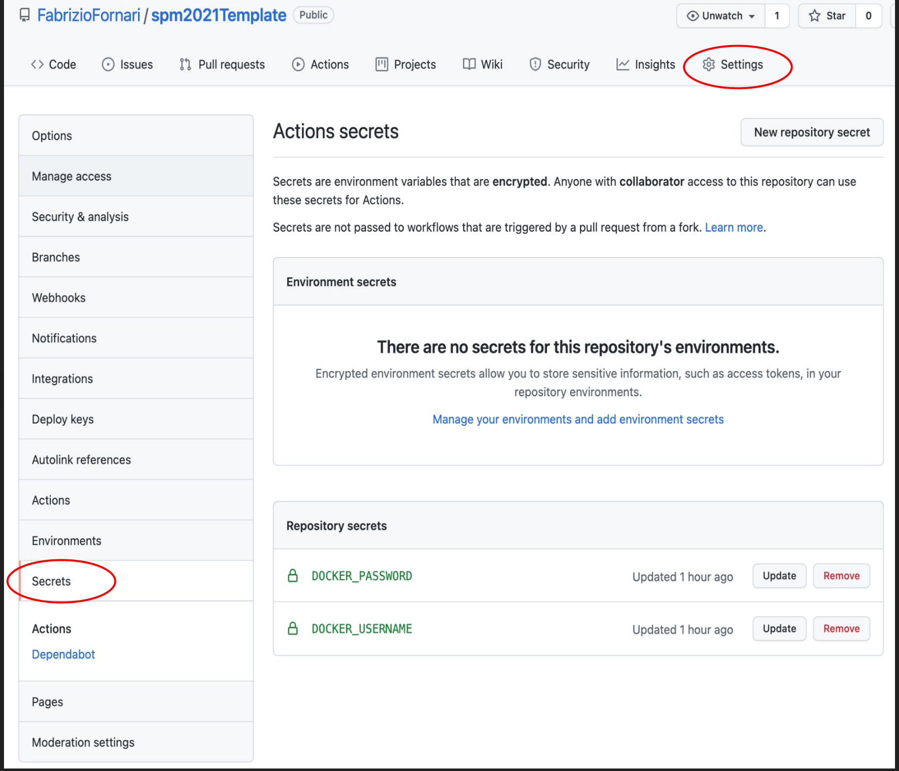
Create an account and a Private Repository


docker hub
Search for great content (e.g., mysql)
Explore
Repositories
Organizations
Help
Search by repository name
Create Repository

fabriziofornari89

GitHub Actions

```
…
   - name: Build with Maven
     run: mvn -B package --file pom.xml test

   - name: Build and Push Docker Image
     uses: mr-smithers-excellent/docker-build-push@v5
     with:
       image: fabriziofornari89/spm2021template
       registry: docker.io
       username: ${{ secrets.DOCKER_USERNAME }}
       password: ${{ secrets.DOCKER_PASSWORD }}
```



Actions secrets

Secrets are environment variables that are **encrypted**. Anyone with **collaborator** access to this repository can use these secrets for Actions.

Secrets are not passed to workflows that are triggered by a pull request from a fork. Learn more.

Environment secrets

There are no secrets for this repository's environments.

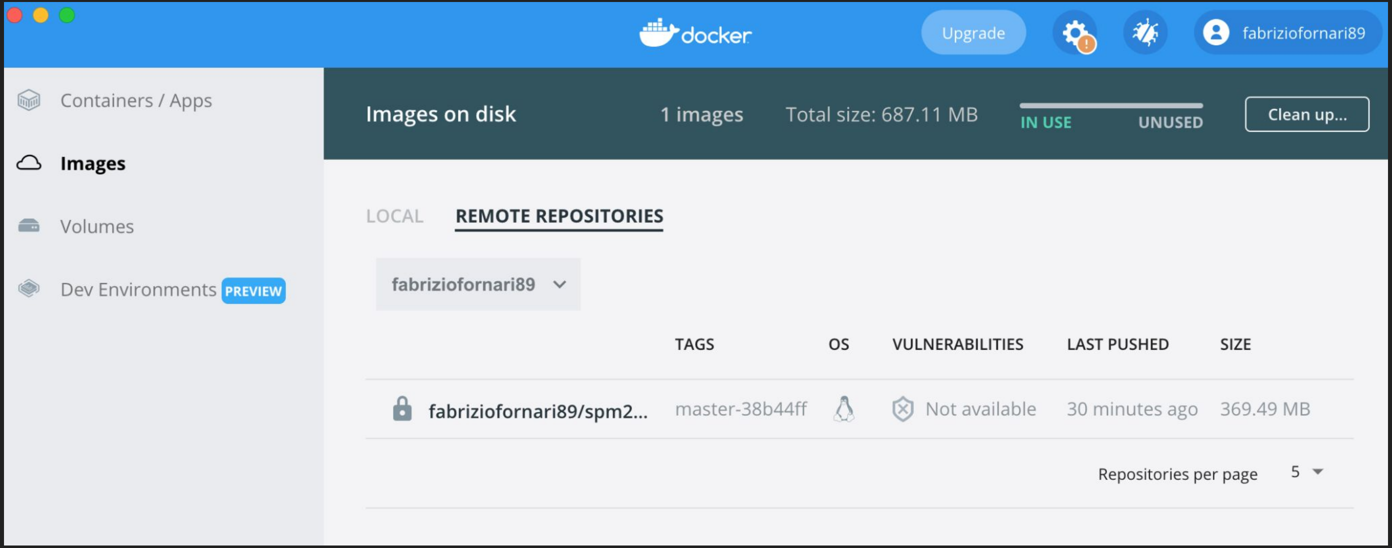
Encrypted environment secrets allow you to store sensitive information, such as access tokens, in your repository environments.

Manage your environments and add environment secrets

Repository secrets

DOCKER_PASSWORD	Updated 1 hour ago
DOCKER_USERNAME	Updated 1 hour ago

Docker Desktop



Images on disk 1 images Total size: 687.11 MB IN USE UNUSED Clean up...

LOCAL REMOTE REPOSITORIES

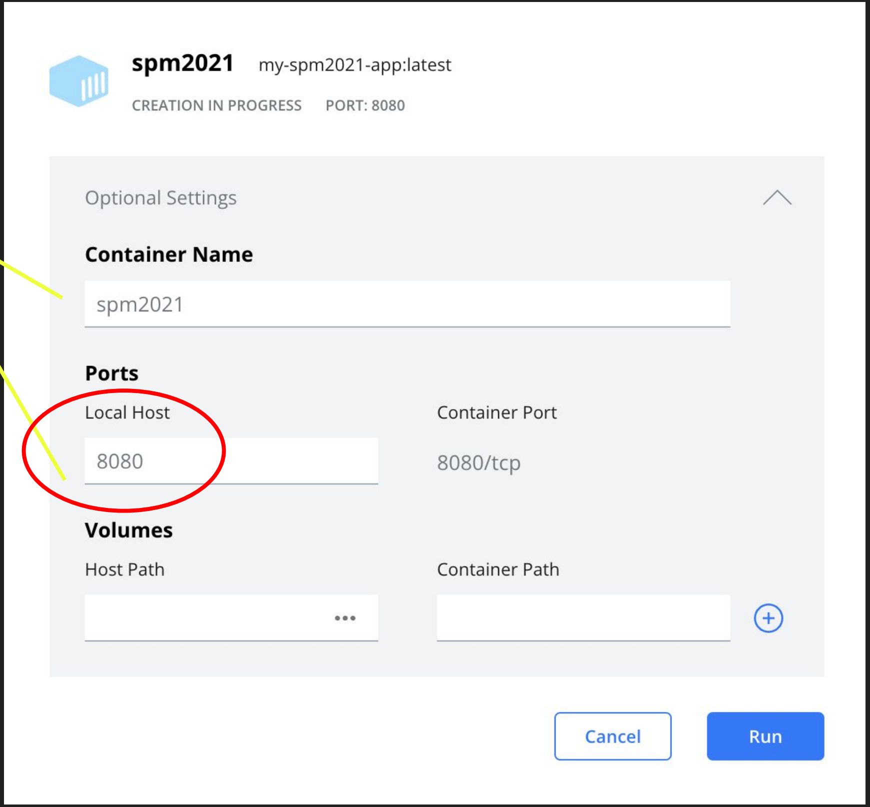
fabriziofornari89

	TAGS	OS	VULNERABILITIES	LAST PUSHED	SIZE
fabriziofornari89/spm2...	master-38b44ff		Not available	30 minutes ago	369.49 MB

Repositories per page 5

Docker Desktop

Setup the Optional Settings so to specify the container name and the host port from which you will access the application



spm2021 my-spm2021-app:latest

CREATION IN PROGRESS PORT: 8080

Optional Settings

Container Name

spm2021

Ports

Local Host: 8080

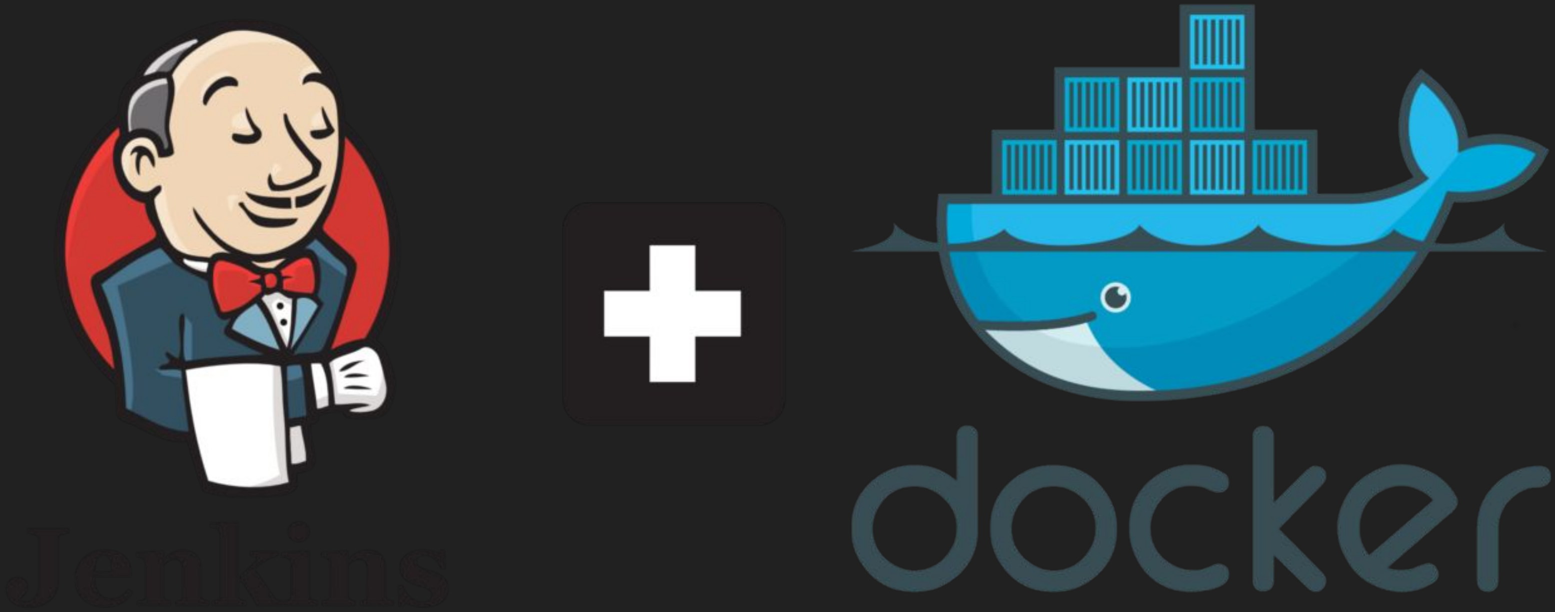
Container Port: 8080/tcp

Volumes

Host Path

Container Path

Jenkins + Docker



...so a Docker Host

What if a Docker Host fails?

Orchestrating Hosts

Orchestration technology focuses on clustering and managing containers and hosts.

Docker Swarm: Easy to setup but lacks autoscaling

Kubernetes: from Google, difficult to setup but supports many advanced features, all public cloud supports it

MESOS: from Apache, difficult to setup but supports many advanced features,

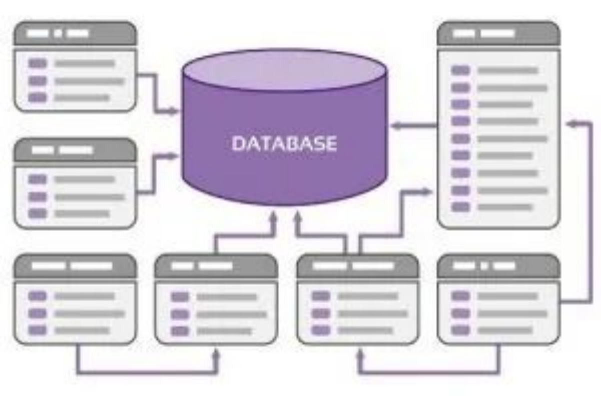
Kubernetes

A fundamental difference between Kubernetes and Docker is that Kubernetes is meant to run across a cluster while Docker runs on a single node. Kubernetes is more extensive than Docker Swarm and is meant to coordinate clusters of nodes at scale in production in an efficient manner.

Lecture 19

Updating a database when working alone is pretty easy.

When working in a team that implements multiple features in parallel, uses different test databases and runs the application on one or more production servers, updating all these databases, keeping track of all executed update operations and merging the changes of your co-workers quickly becomes an issue.



Special Guest

Jasmin Fluri works as an Infrastructure Engineering Consultant at Schaltstelle GmbH in Switzerland and lectures on software engineering at the University of Applied Sciences North-Western Switzerland (FHNW) in Windisch.

Her focus as a consultant lies on CI/CD, building automated pipelines and automation of recurring tasks.

She's currently writing her Master Thesis on the topic of Database Schema Evolution and Testing during Continuous Integration.

How Databases fit into CI/CD?

Especially regarding Relational Database we can speak about Database Migration

Flyway

Version control for your database

Robust schema evolution across all your environments.
With ease, pleasure, and plain SQL.

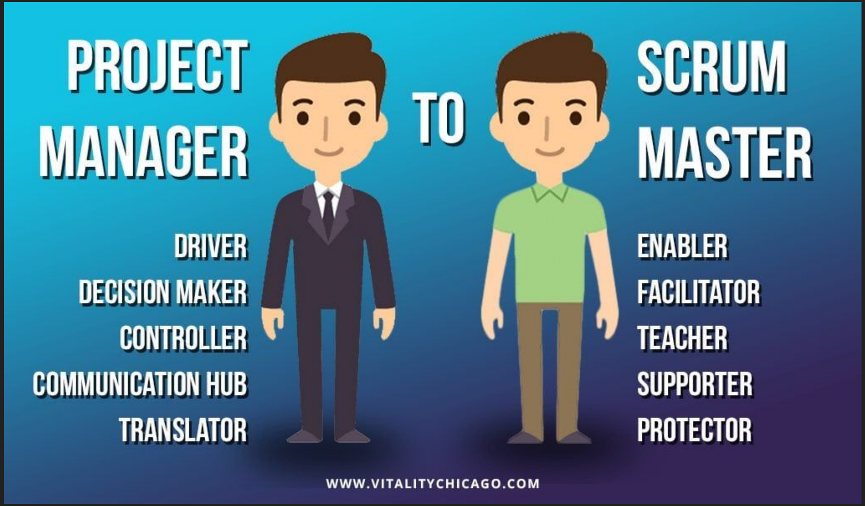
Group Projects or Thesis

I supervise group projects and experimental thesis.

I try to apply together with the students the methodology and tools that we have seen during the course.

You can contact me for any question related to the course and for additional information about projects and thesis: *fabrizio.fornari@unicam.it*

Note: only email coming from the @studenti.unicam.it domain will be processed.



Fill the evaluation questionnaire

https://www.unicam.it/studente/questionari-sulla-didattica