Software Project Management - Laboratory

Lecture n° 21
A.Y. 2020-2021

Prof. Fabrizio Fornari
fabrizio.fornari@unicam.it

Questionario Online

http://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

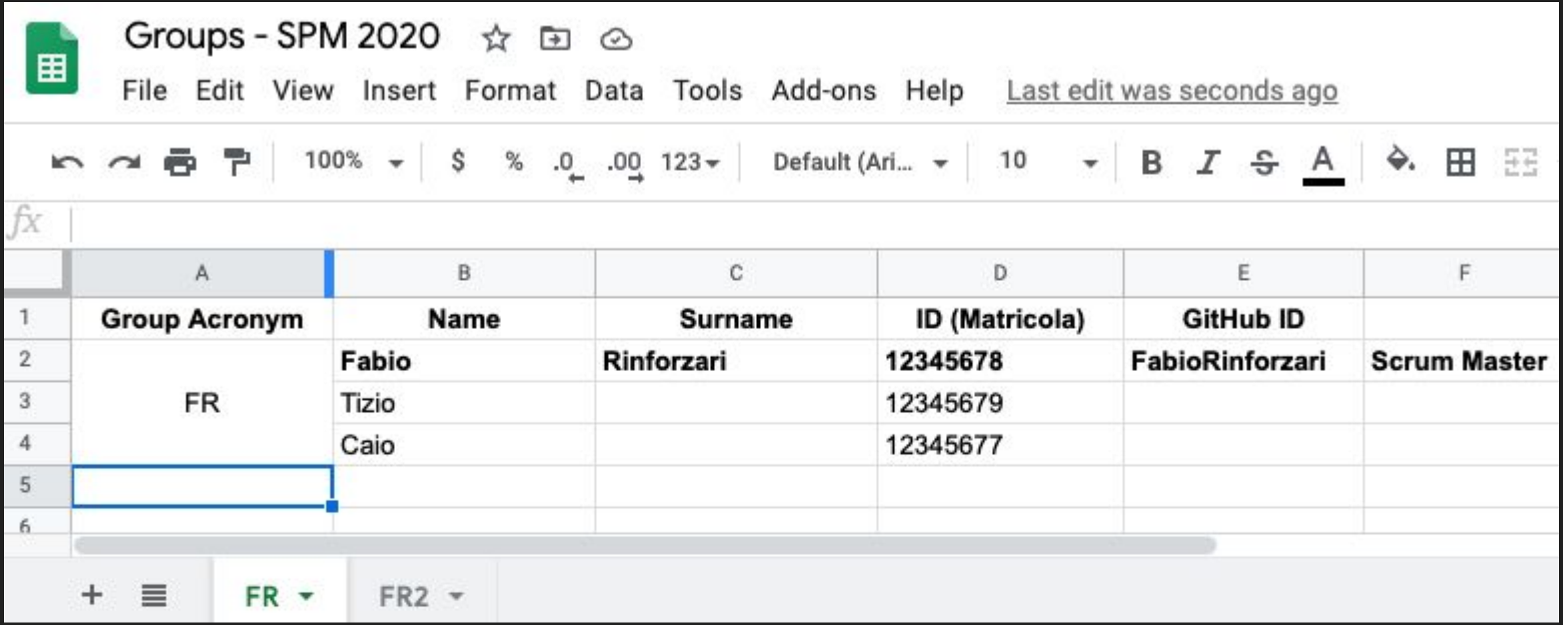
Learning Outcome

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

Google Sheets

https://docs.google.com/spreadsheets/d/1tOdMoBKzBjcHGQI3ACmrkkcaFbc2TuPfgmFQM2jL9BU

https://tinyurl.com/yynuktcl

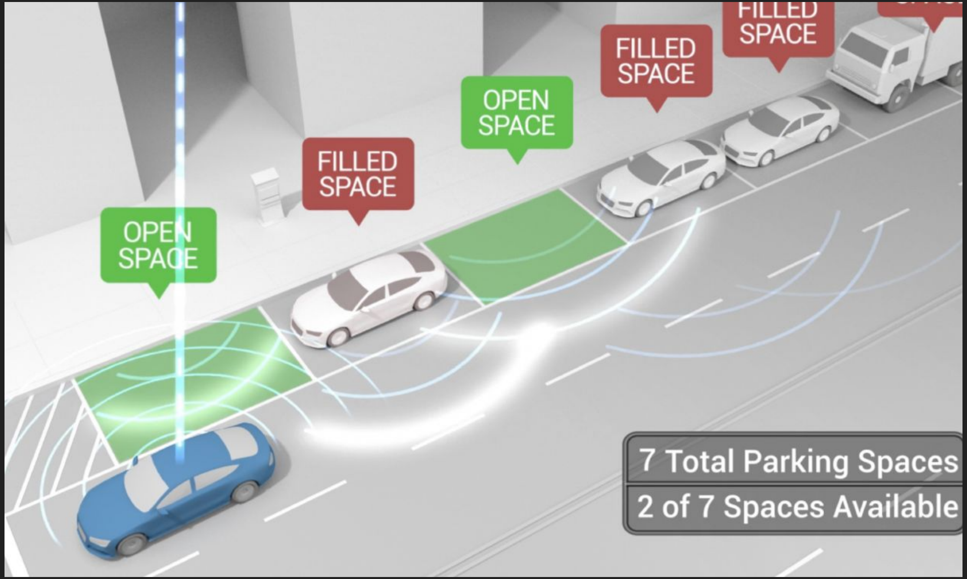

Groups - SPM 2020
File Edit View Insert Format Data Tools Add-ons Help
Last edit was seconds ago
FR
FR2

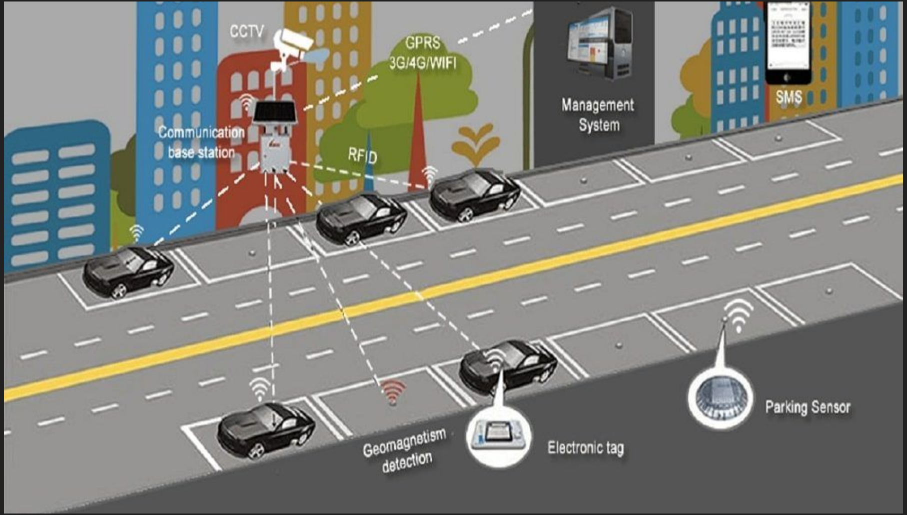
	A	B	C	D	E	F
1	Group Acronym	Name	Surname	ID (Matricola)	GitHub ID	
2		Fabio	Rinforzari	12345678	FabioRinforzari	Scrum Master
3	FR	Tizio		12345679		
4		Caio		12345677		
5						
6						

Smart Parking

Actors:

- Driver
- Municipality
- Policeman
- Tow Truck
- ...





Scenarios:

- Find a Parking Spot
- Drive to the Parking Spot
- Pay the Parking Spot
- Check the Parking Spot has been paid
- Change Parking Spot Status (pay per use vs free parking spot)
- Analyse Parking Status History
- ...

Smart Parking

https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=smart+parking

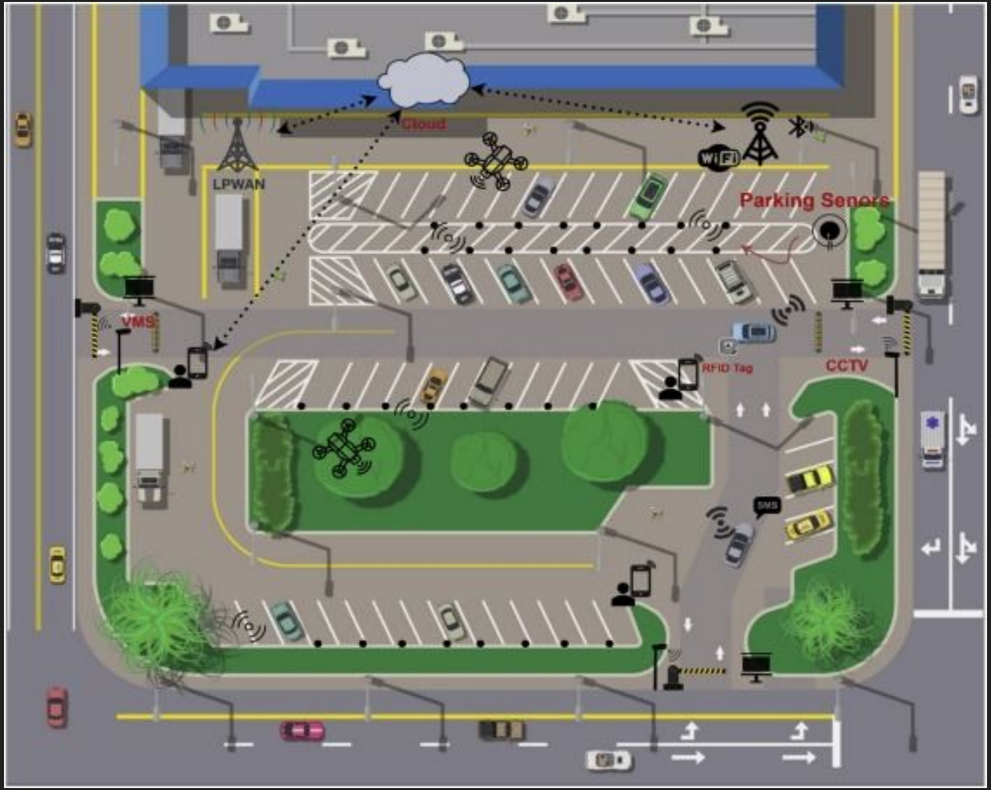
Some Literature:

- Al-Turjman, F., & Malekloo, A. (2019). *Smart parking in IoT-enabled cities: A survey. Sustainable Cities and Society*, 49, 101608.

- Lin, T., Rivano, H., & Le Mouël, F. (2017). *A survey of smart parking solutions*. IEEE Transactions on Intelligent Transportation Systems, 18(12), 3229-3253.

- Saleem, Y., Sotres, P., Fricker, S., de la Torre, C. L., Crespi, N., Lee, G. M., ... & Sanchez, L. (2020). IoTRec: *The IoT Recommender for Smart Parking System*. IEEE Transactions on Emerging Topics in Computing.

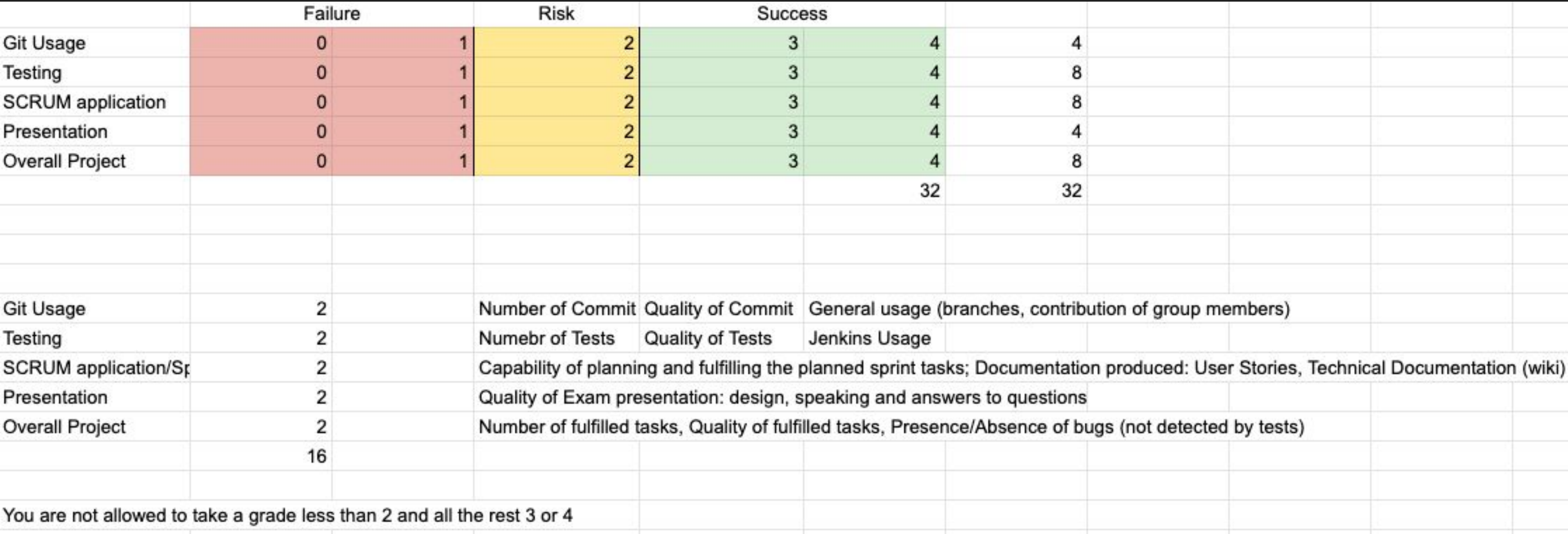
- Lin, T., Rivano, H., & Le Mouël, F. (2017). *A survey of smart parking solutions*. IEEE Transactions on Intelligent Transportation Systems, 18(12), 3229-3253.



Exam Evaluation

Evaluation Schema

https://docs.google.com/spreadsheets/d/16D7shtTEO1BNxeauj_X5H25wYkG8DQQGYHenOeD1KOM/edit?usp=sharing



	Failure		Risk	Success		
Git Usage	0	1	2	3	4	4
Testing	0	1	2	3	4	8
SCRUM application	0	1	2	3	4	8
Presentation	0	1	2	3	4	4
Overall Project	0	1	2	3	4	8
					32	32

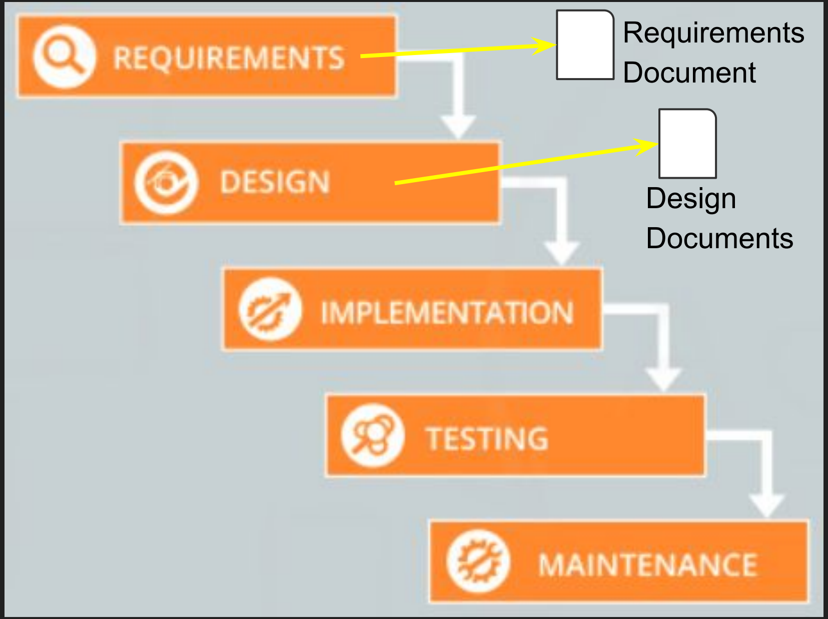
Git Usage	2		Number of Commit	Quality of Commit	General usage (branches, contribution of group members)
Testing	2		Numebr of Tests	Quality of Tests	Jenkins Usage
SCRUM application/Sp	2		Capability of planning and fulfilling the planned sprint tasks; Documentation produced: User Stories, Technical Documentation (wiki)		
Presentation	2		Quality of Exam presentation: design, speaking and answers to questions		
Overall Project	2		Number of fulfilled tasks, Quality of fulfilled tasks, Presence/Absence of bugs (not detected by tests)		
	16				

You are not allowed to take a grade less than 2 and all the rest 3 or 4

Software Development Process

Software Development Process is the process of dividing software development work into distinct phases to improve design, product management, and project management. It is also known as a software development life cycle (SDLC)

Waterfall Development Process



The Waterfall Model is document driven.

Each step of the process yields artifacts that are documents.

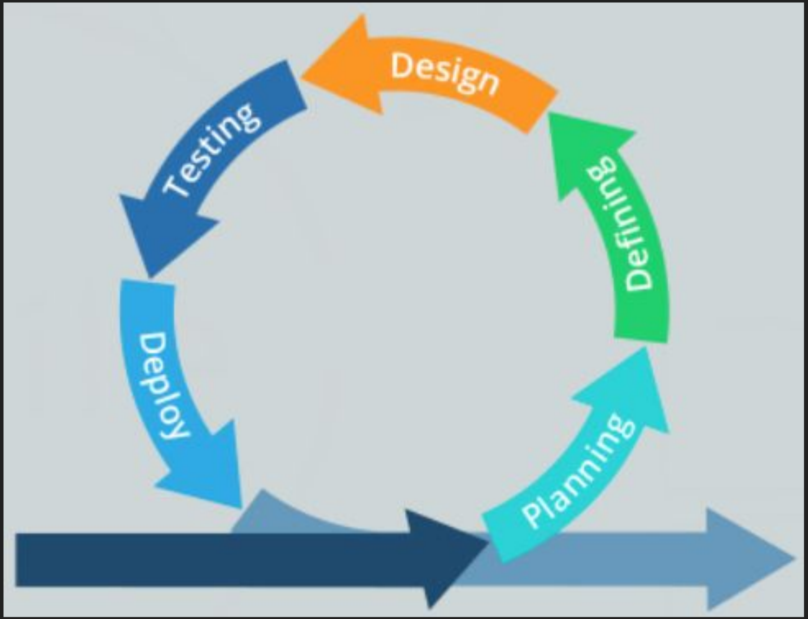
Documents produced during one step are needed for the next step and possibly for later steps.

Managers like the waterfall model because progress is observable and measurable.

Knowledge is lost if team members leave before the project is completed, and it may be difficult for a project to recover from the loss. That is the reason why documents are important.

However, it has repeatedly proven to be very easy to produce impressive documents that eventually prove to be incomplete, inconsistent, hard to consult, or otherwise worthless.

Agile Development Process



Manifesto for Agile Software Development

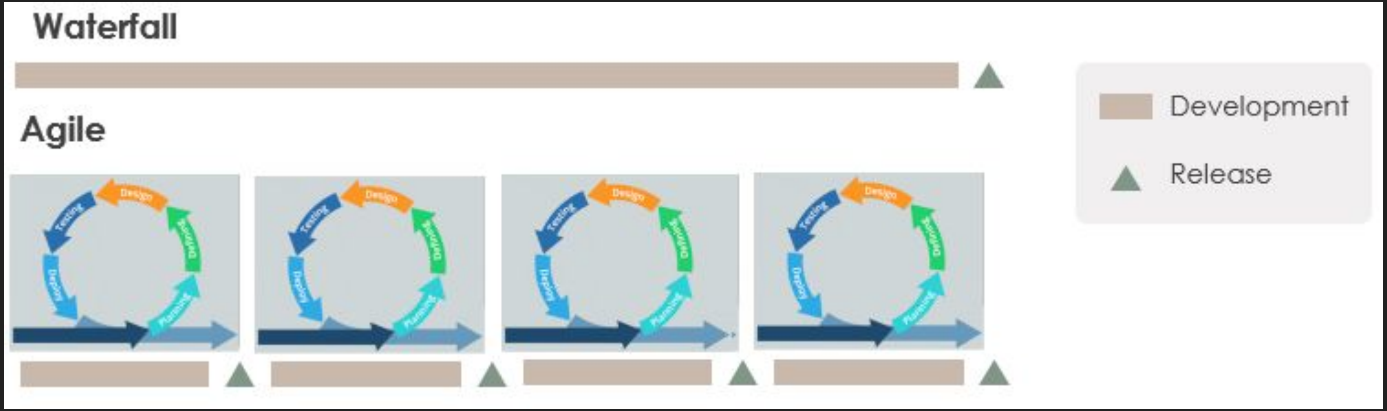
Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

Waterfall vs Agile



Agile characteristics:

- Complex work is divided in simple pieces
- Large organizations are divided into small teams
- Far-reaching project are divided into short time lists of task called sprints

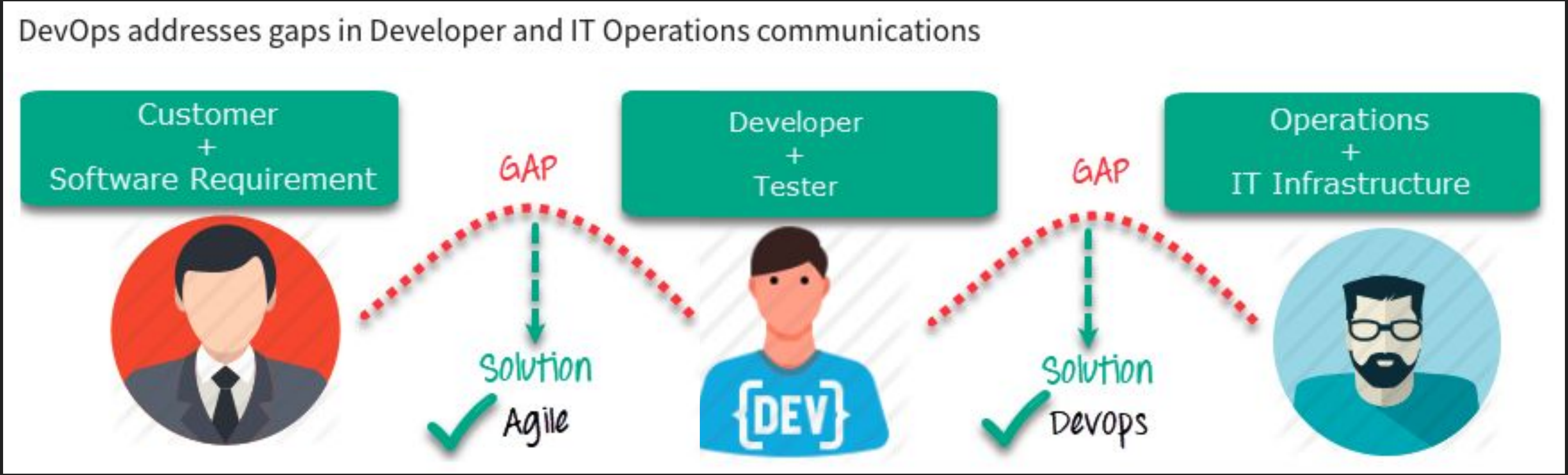
DevOps

“A compound of development (Dev) and operations (Ops), DevOps is the union of people, process, and technology to continually provide value to customers.

What does DevOps mean for teams? DevOps enables formerly siloed roles — development, IT operations, quality engineering, and security — to coordinate and collaborate to produce better, more reliable products. By adopting a DevOps culture along with DevOps practices and tools, teams gain the ability to better respond to customer needs, increase confidence in the applications they build, and achieve business goals faster.”

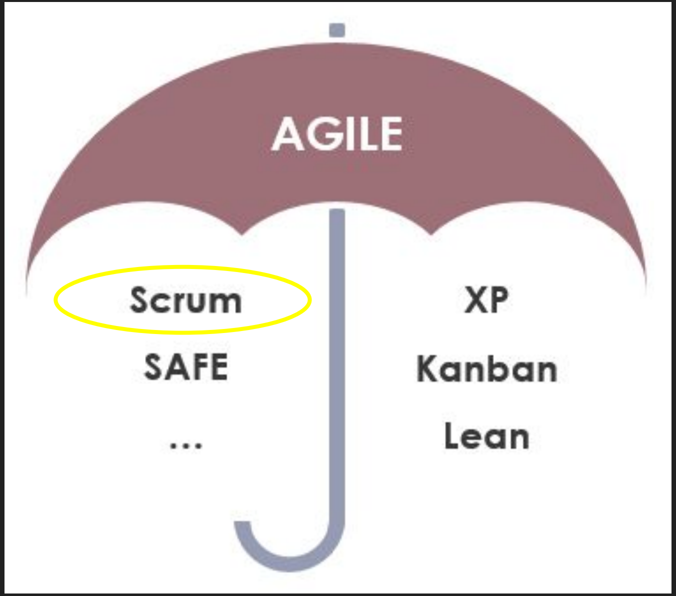
— Azure.microsoft.com

DevOps


DevOps addresses gaps in Developer and IT Operations communications
Customer + Software Requirement
GAP
Solution
Agile
Developer + Tester
{DEV}
GAP
Solution
Devops
Operations + IT Infrastructure

SCRUM

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



Roles

Product Owner - is responsible for working with the user group to determine what features will be in the product release. Some of the responsibilities:

- Develop the direction and strategy for the products and services, including the short and long-time goals;
- Provide or have access to knowledge about the product or the service;
- Understand and explain customer needs for the Development team;

Scrum Master - is the facilitator for an agile development team. Some of the responsibilities:

- Act as a coach, helping the team to follow scrum values and practices;
- Help to remove impediments and protect the team from external interferences;
- Promote a good cooperation between the team and stakeholders;

Scrum Team - is formed by 3 to 9 people who MUST fulfill all technical needs to deliver the product or the service. They will be guided directly by the Scrum Master, but they will not be directly managed. They must be self-organized, versatile, and responsible enough to complete all required tasks.

Artifacts

The SCRUM artifacts are used to help define the workload coming into the team and currently being worked upon the team.

The main artifacts:

- **Product backlog** - a collection of user stories which present functionalities required/wanted by the product team. Usually the product owner takes responsible for this list.

- **Sprint backlog** - a collection of stories which could be included in the current sprint.

User Stories

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.

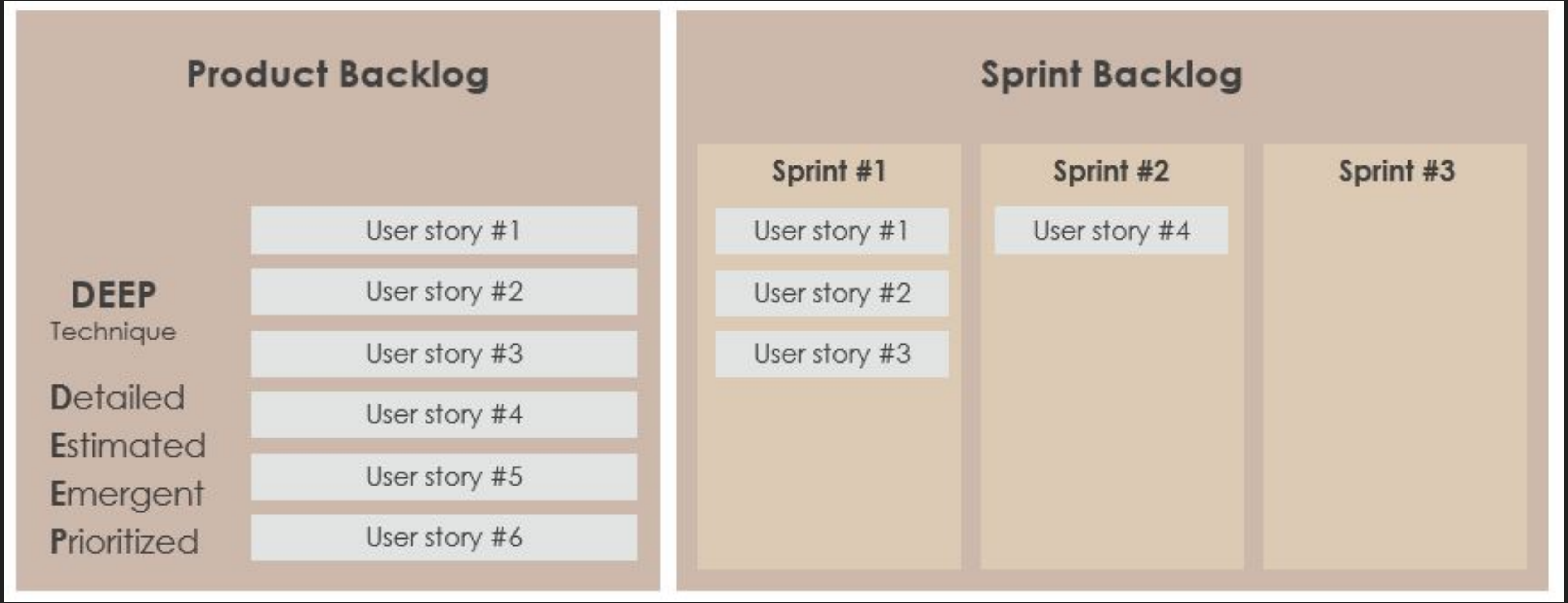
Can generally follow a template:

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).

Artifacts: Product Backlog and Sprint Backlog



Sprint

In the Scrum Framework all activities needed for the implementation of entries from the Scrum Product Backlog are performed within Sprints (also called ‘Iterations’). Sprints are always short: normally about 2-4 weeks.

Epics > User Stories > Tasks

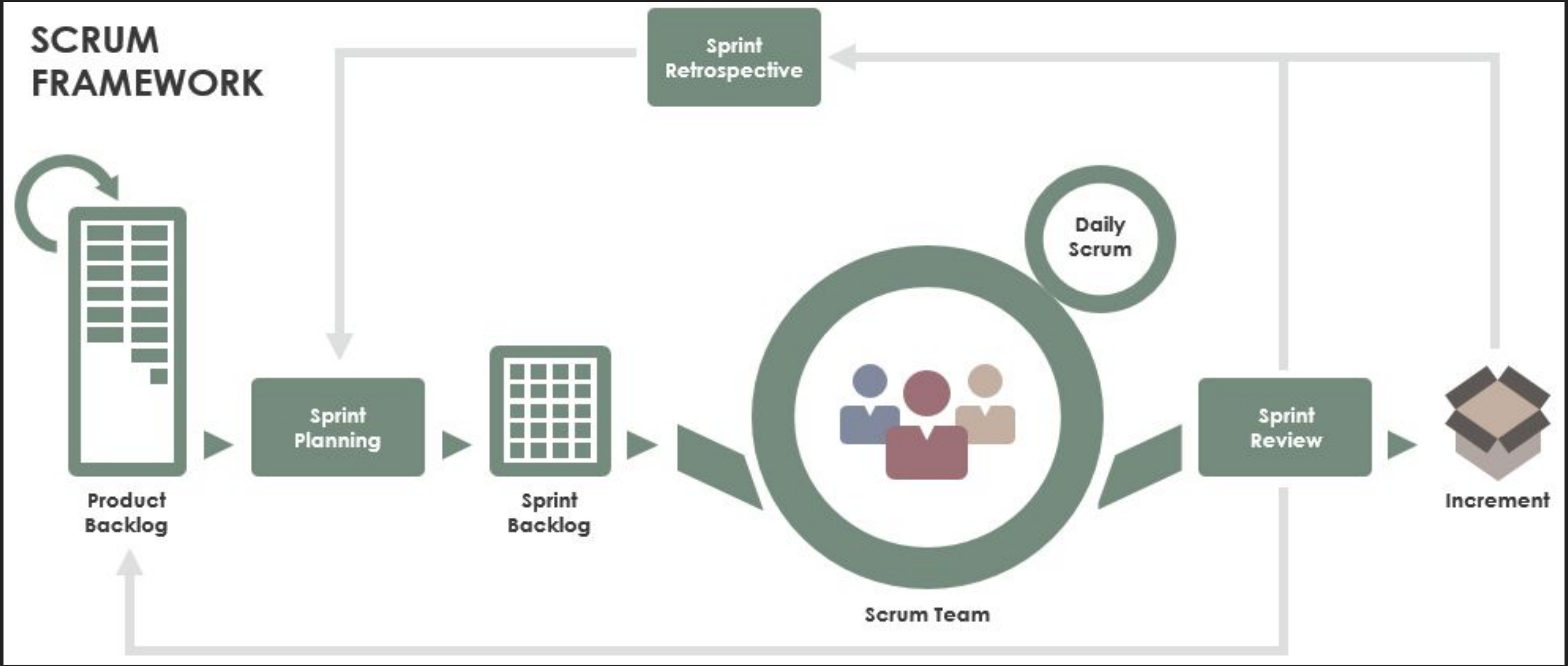
An epic captures a large body of work. It is essentially a “large user story” that can be broken down into a number of smaller stories.

A story is a brief statement of a product requirement or a business case.

A task is typically something like “code this”, “design that”, “create test data for such-and-such”, and so on. Tend to be things done by one person. A task is not written in the user story format. A task has more a technical nature.

Experienced Scrum Teams spend time and effort to break down complex and larger items (i.e user features or epics) into smaller user stories (or subsequently breaking down into tasks, or subtasks).

SCRUM


SCRUM FRAMEWORK
Sprint Retrospective
Daily Scrum
Sprint Planning
Sprint Review
Product Backlog
Sprint Backlog
Increment
Scrum Team

Additional Materials

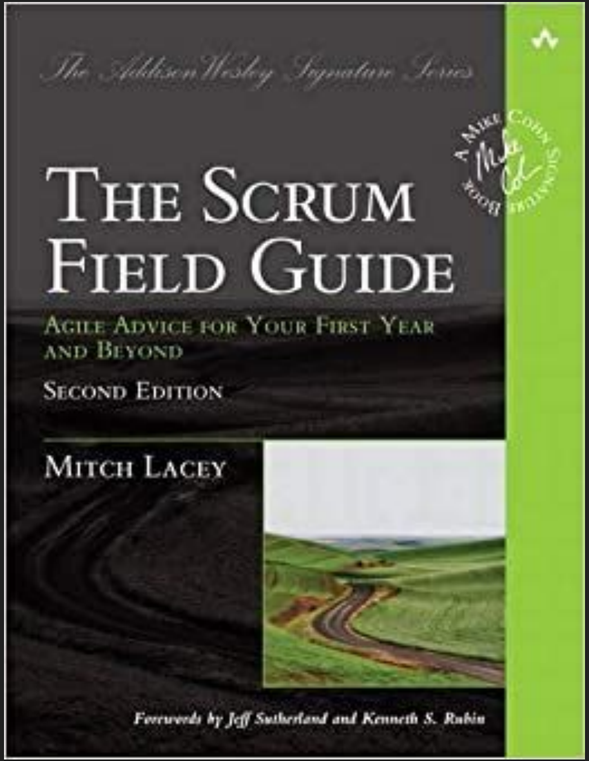
Scrum Field Guide,

The: Agile Advice for Your First Year and Beyond (Addison-Wesley Signature Series (Cohn)) 2nd Edition

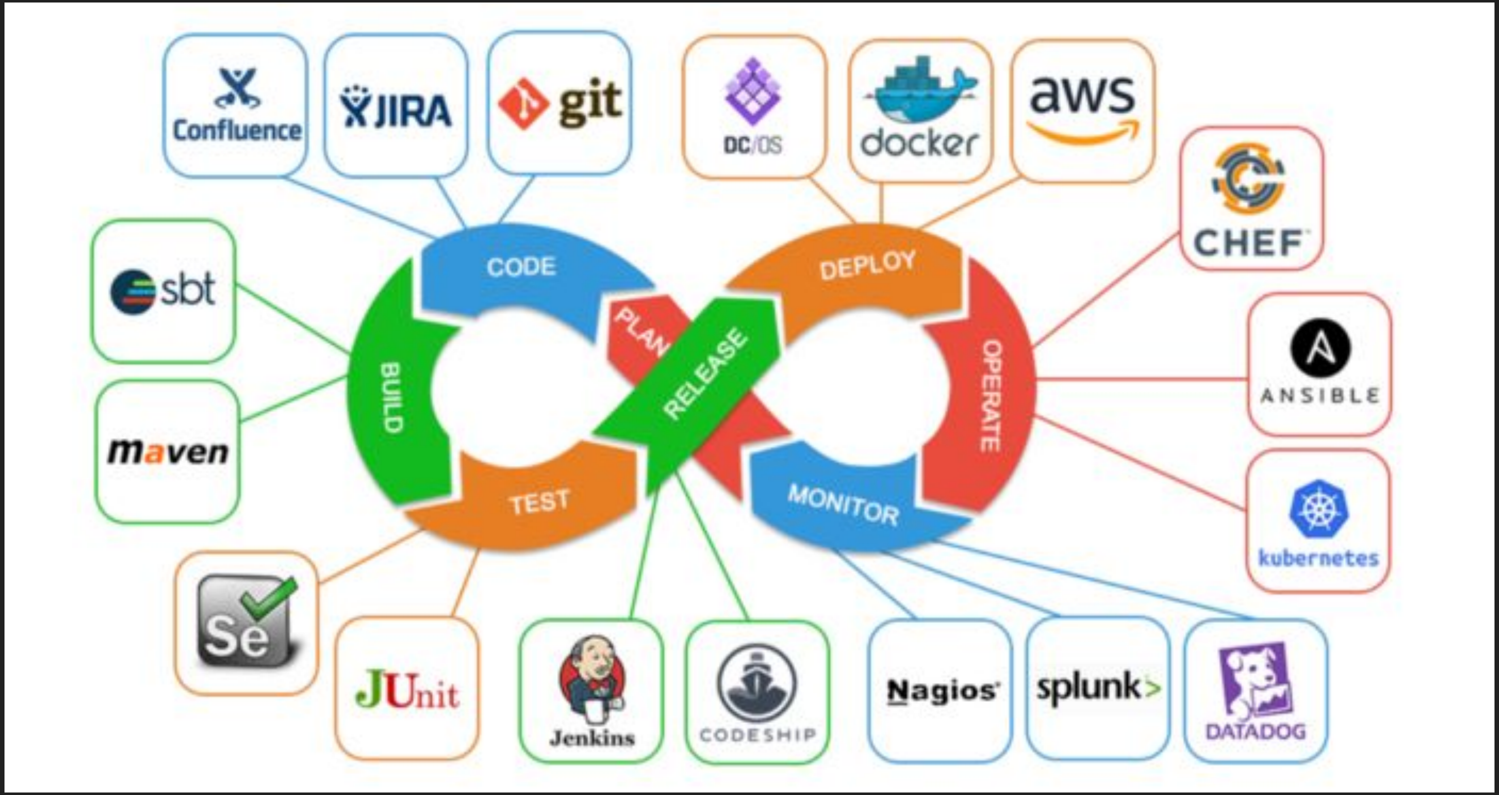
By Mitch Lacey

https://www.mountaingoatsoftware.com/

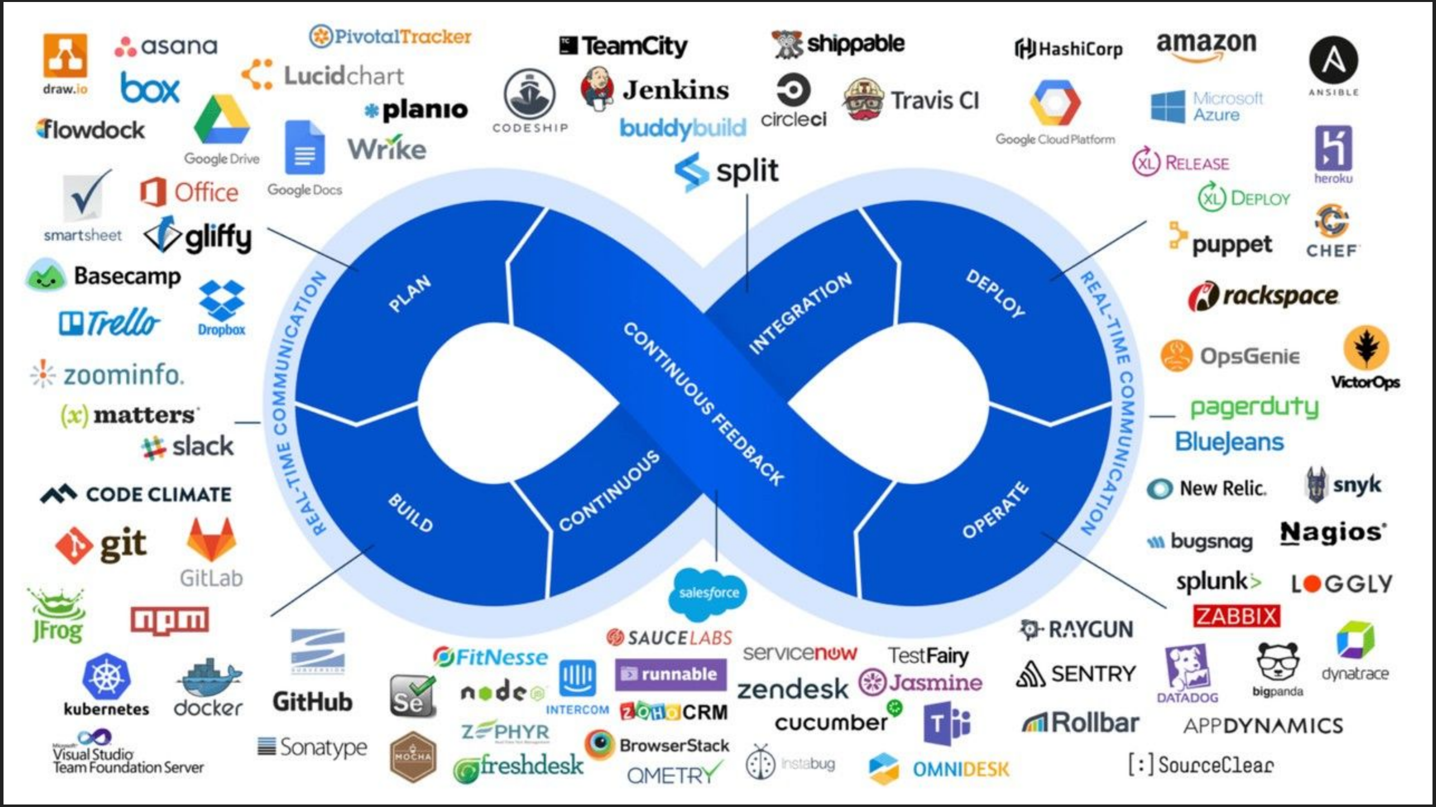
By Mike Cohn


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

DevOps Technologies



DevOps Technologies

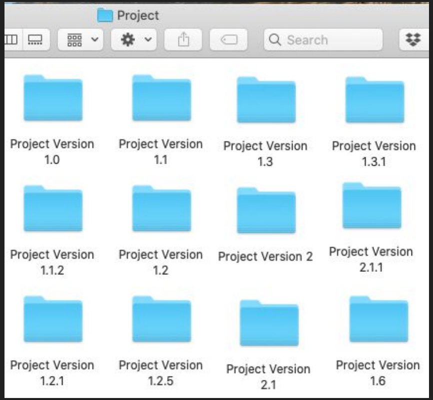


Version Control

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

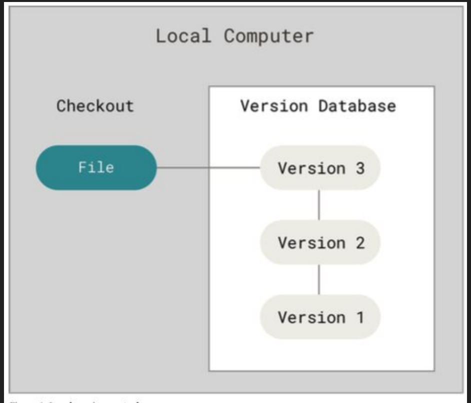


Figure 1. Local version control

Distributed Version Control Systems

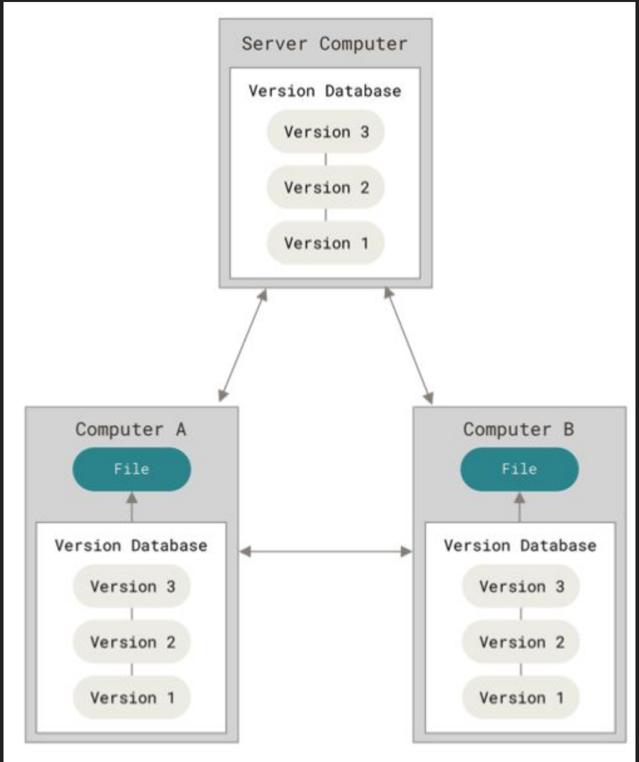


Figure 3. Distributed version control

Centralized Version Control Systems

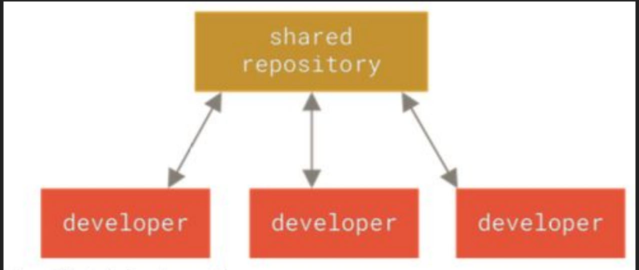
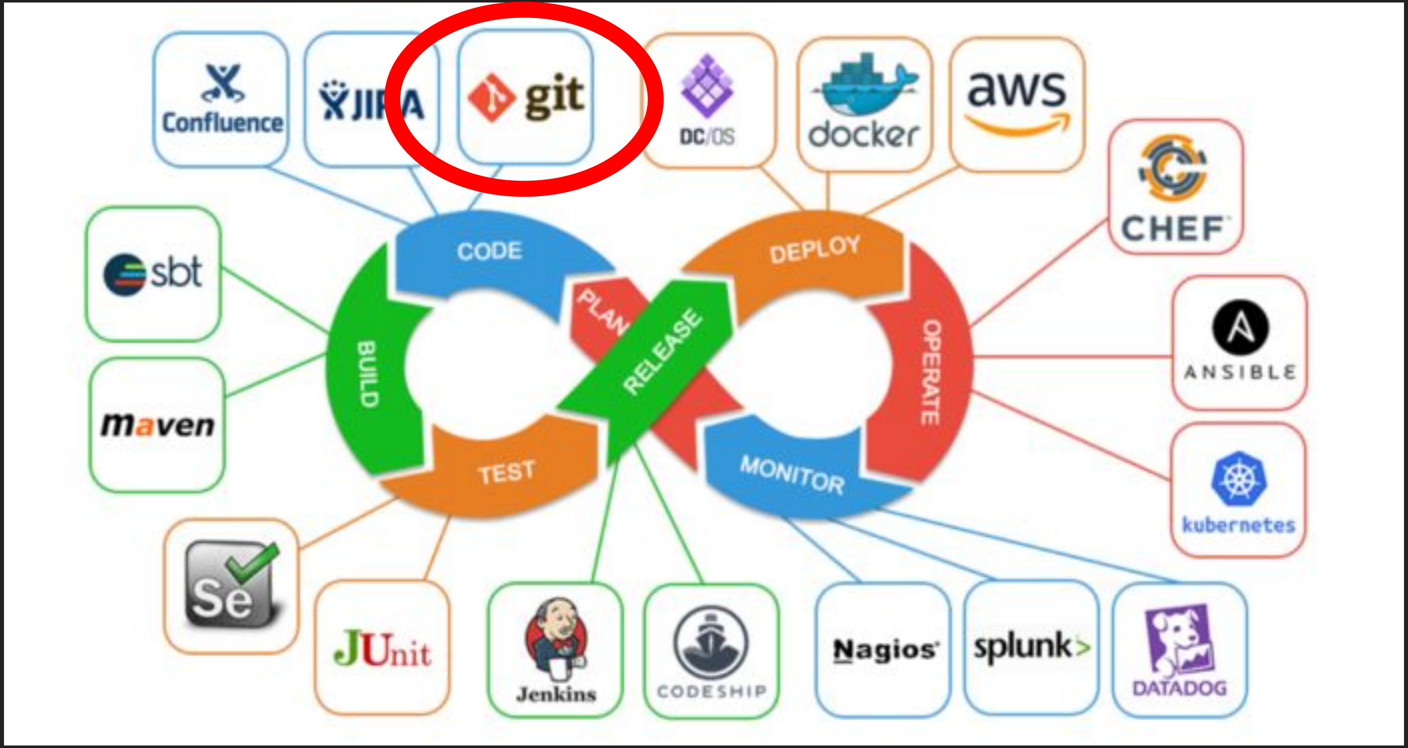


Figure 2. Centralized version control

DevOps


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

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.

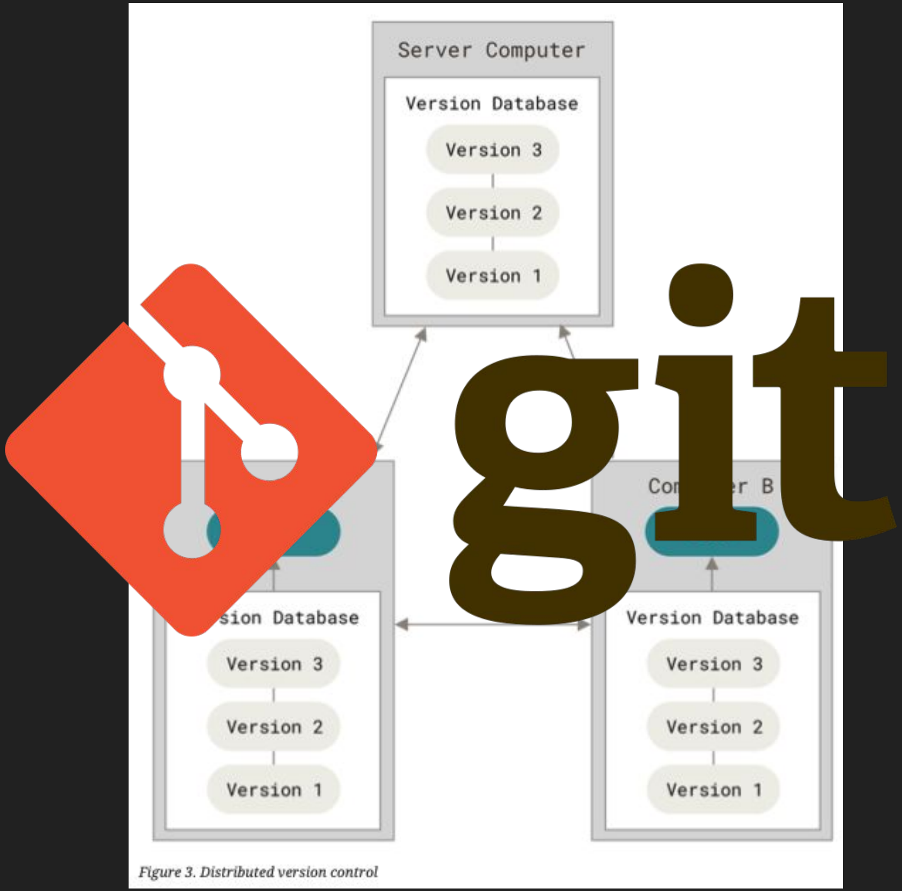


Figure 3. Distributed version control

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

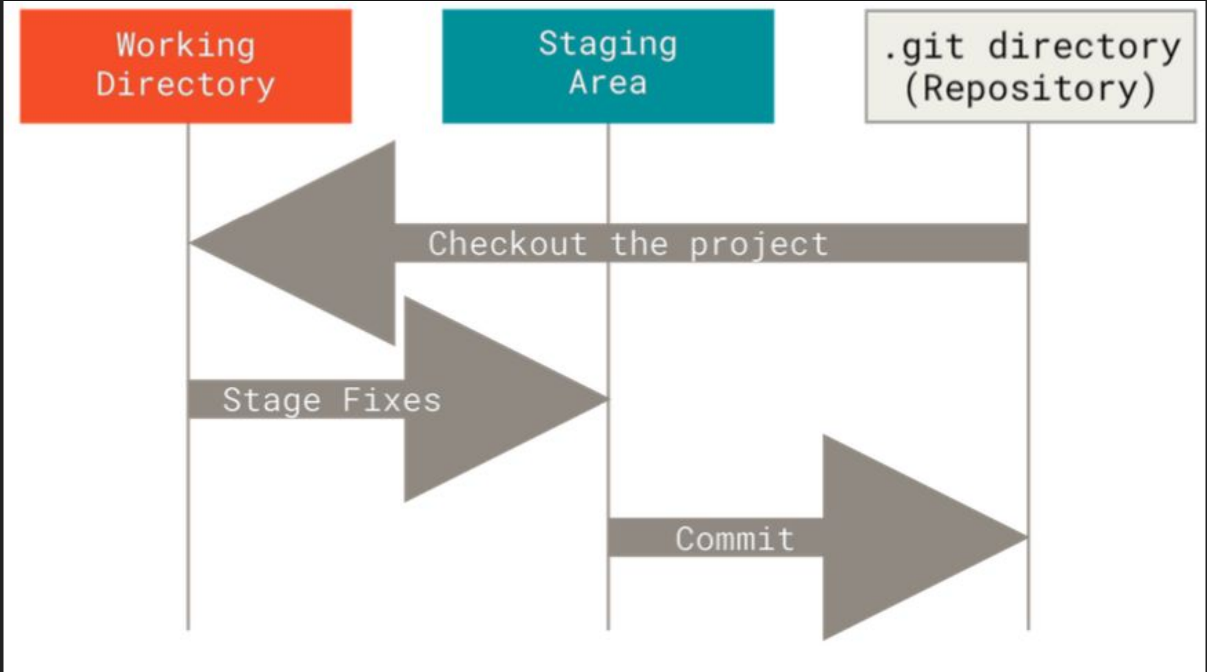
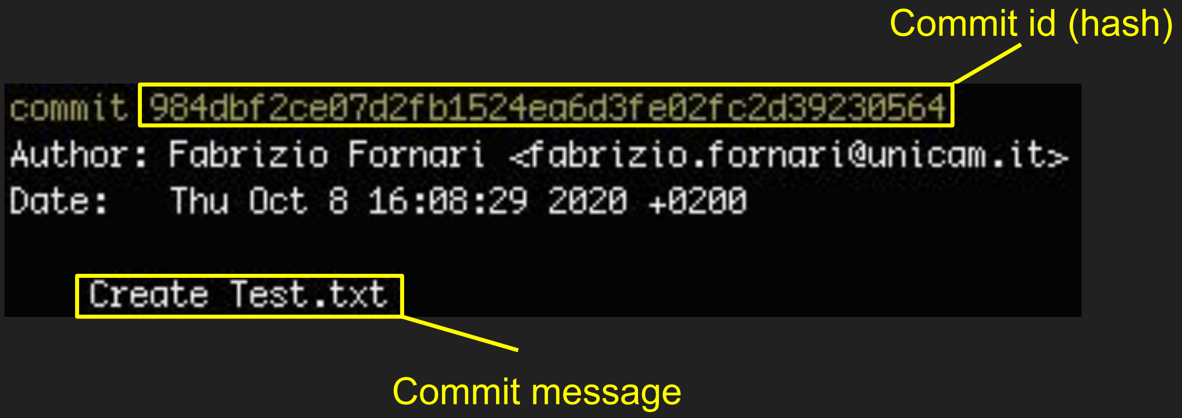


Figure 6. Working tree, staging area, and Git directory

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
```

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.

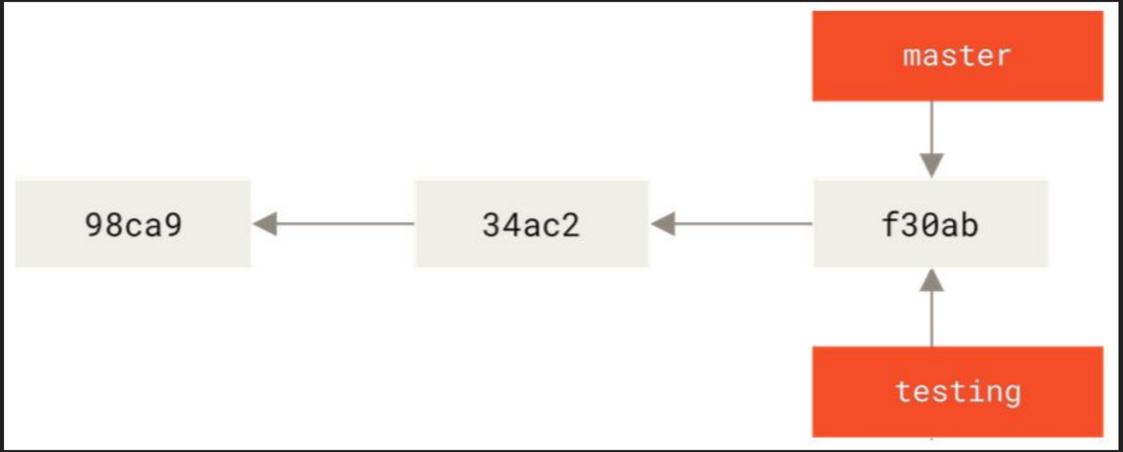
```
$ git config --global user.name "Name Surname"

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

4. You can check your settings at any time:

```
$git config −−l i s t
```

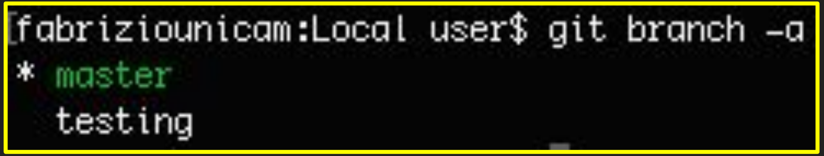
Creating a New Branch

1. Run git branch testing
2. Run git status



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

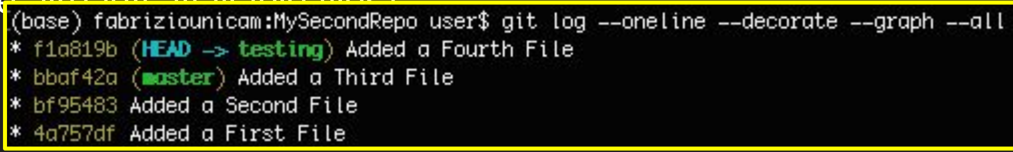
3. Run git branch -a



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

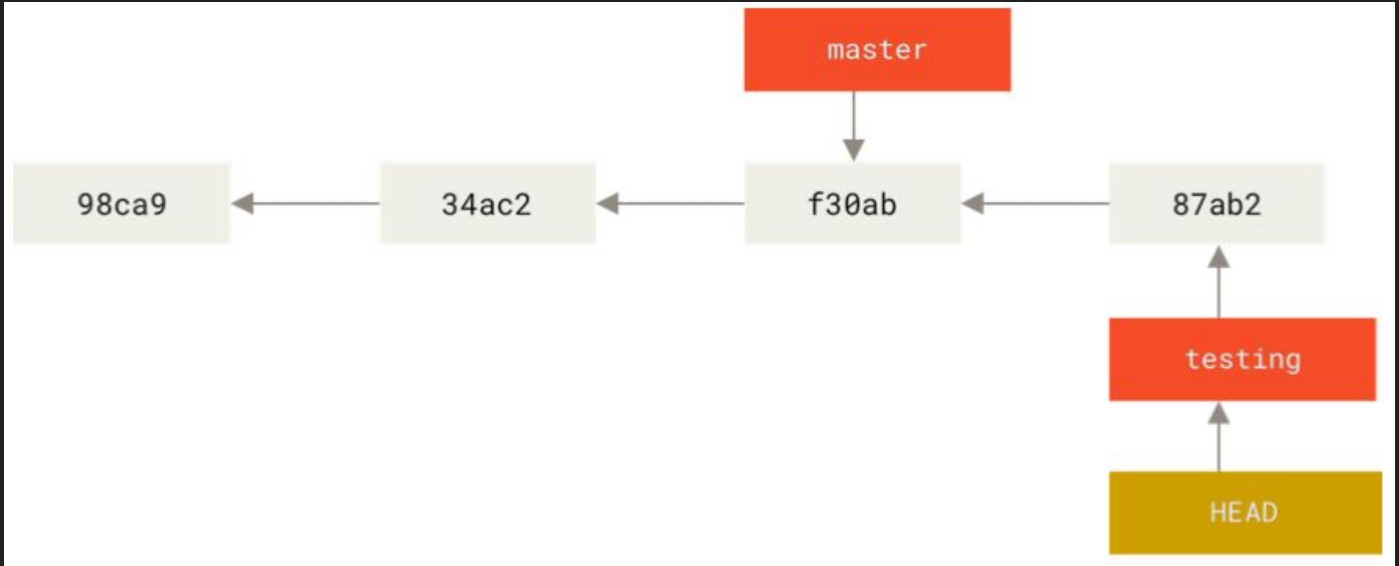
Commit to a New Branch

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



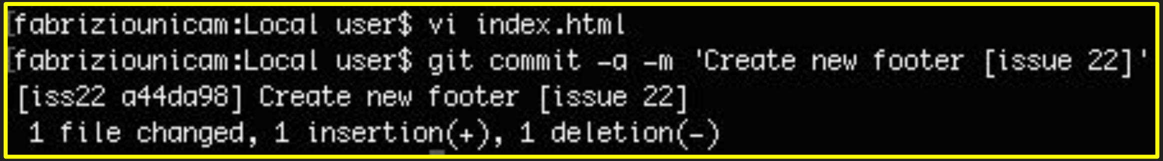
```
(base) fabriziounicam:MySecondRepo user$ git log --oneline --decorate --graph --all
* f1a819b (HEAD -> testing) Added a Fourth File
* bbaf42a (master) Added a Third File
* bf95483 Added a Second File
* 4a757df Added a First File
```

Your testing branch has moved forward



Branching and Merging

You do some changes and you commit

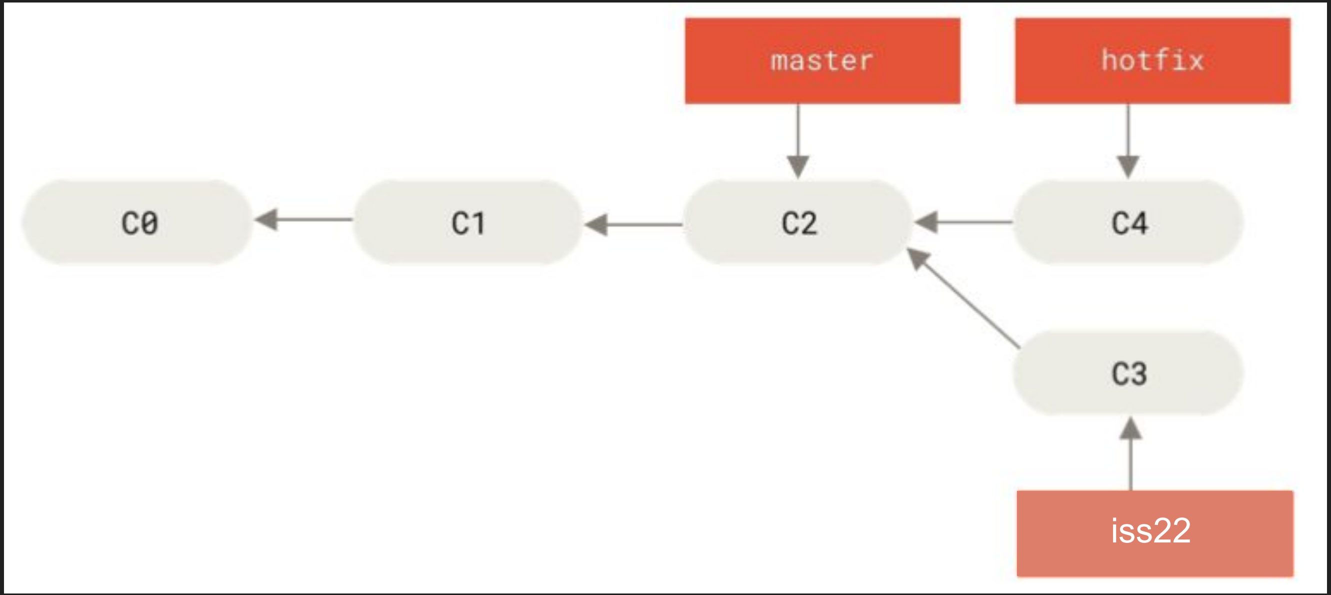


```
fabriziounicam:Local user$ vi index.html
fabriziounicam:Local user$ git commit -a -m 'Create new footer [issue 22]'
[iss22 a44da98] Create new footer [issue 22]
 1 file changed, 1 insertion(+), 1 deletion(-)
```

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

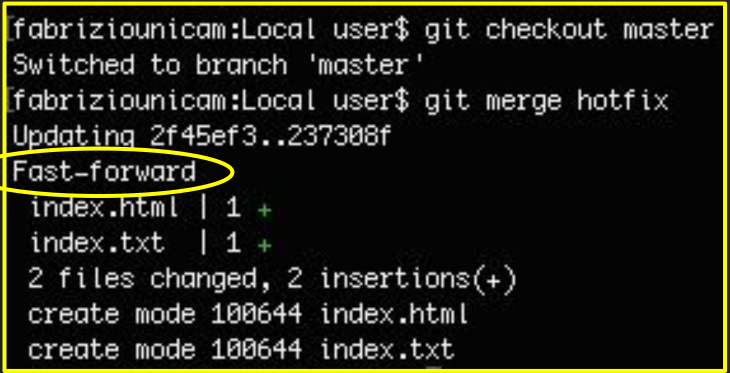


Branching and Merging

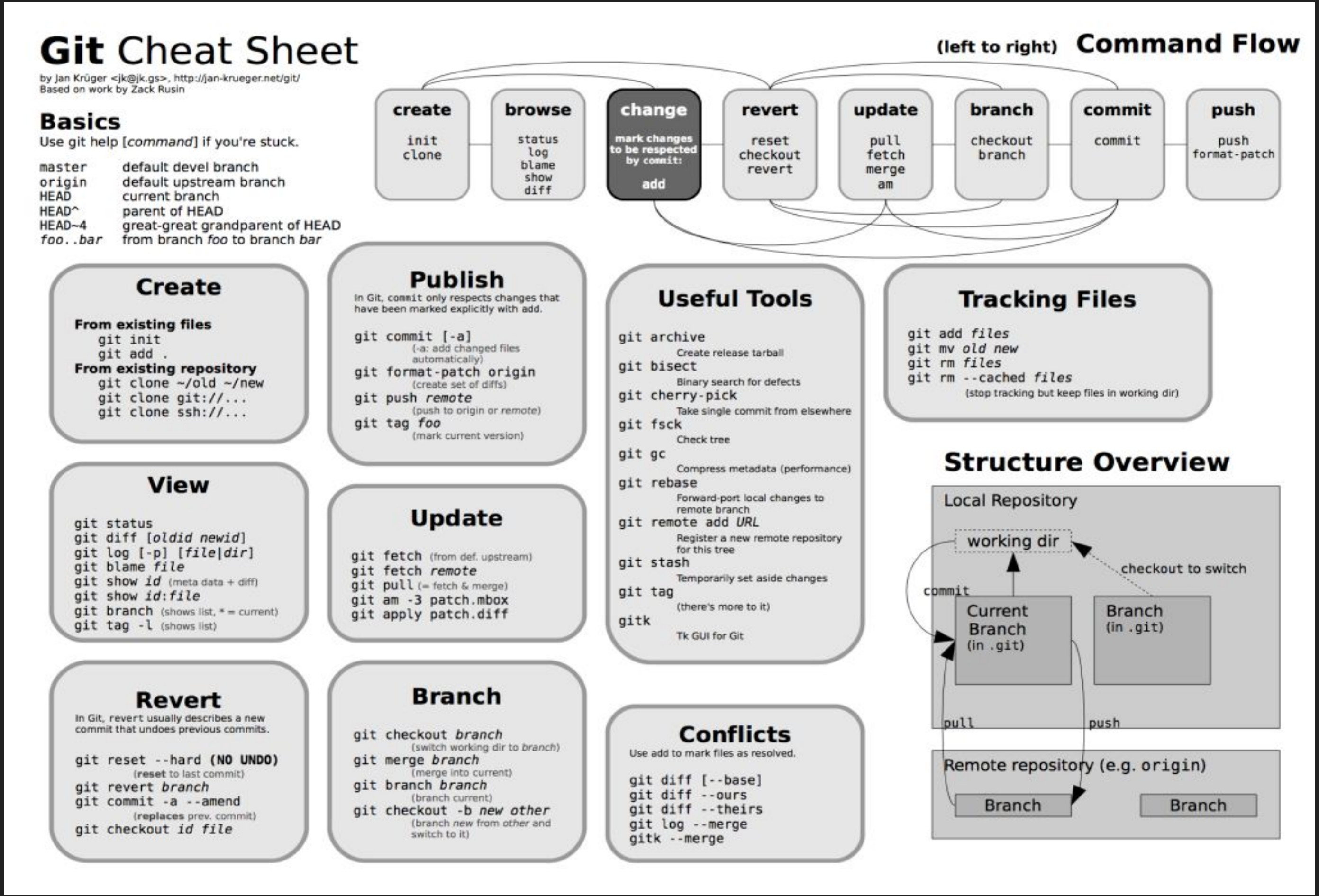
You can run your tests, make sure the hotfix is what you want, and finally merge the hotfix branch back into your master branch to deploy to production.

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
```



Git Cheat Sheet

by Jan Krüger <jk@jk.gs>, http://jan-krueger.net/git/
Based on work by Zack Rusin

(left to right) **Command Flow**

create	browse	change	revert	update	branch	commit	push
init clone	status log blame show diff	mark changes to be respected by commit: add	reset checkout revert	pull fetch merge am	checkout branch	commit	push format-patch

Basics

Use git help [*command*] if you're stuck.

`master`	default devel branch
`origin`	default upstream branch
`HEAD`	current branch
`HEAD^`	parent of HEAD
`HEAD~4`	great-great grandparent of HEAD
`foo..bar`	from branch *foo* to branch *bar*

Create

From existing files

```
git init
git add .
```

From existing repository

```
git clone ~/old ~/new
git clone git://...
git clone ssh://...
```

View

```
git status
git diff [oldid newid]
git log [-p] [file|dir]
git blame file
git show id (meta data + diff)
git show id:file
git branch (shows list, * = current)
git tag -l (shows list)
```

Revert

In Git, revert usually describes a new commit that undoes previous commits.

```
git reset --hard (NO UNDO)
    (reset to last commit)
git revert branch
git commit -a --amend
    (replaces prev. commit)
git checkout id file
```

Publish

In Git, commit only respects changes that have been marked explicitly with add.

```
git commit [-a]
    (-a: add changed files automatically)
git format-patch origin
    (create set of diffs)
git push remote
    (push to origin or remote)
git tag foo
    (mark current version)
```

Update

```
git fetch (from def. upstream)
git fetch remote
git pull (= fetch & merge)
git am -3 patch.mbox
git apply patch.diff
```

Branch

```
git checkout branch
    (switch working dir to branch)
git merge branch
    (merge into current)
git branch branch
    (branch current)
git checkout -b new other
    (branch new from other and switch to it)
```

Useful Tools

```
git archive
    Create release tarball
git bisect
    Binary search for defects
git cherry-pick
    Take single commit from elsewhere
git fsck
    Check tree
git gc
    Compress metadata (performance)
git rebase
    Forward-port local changes to remote branch
git remote add URL
    Register a new remote repository for this tree
git stash
    Temporarily set aside changes
git tag
    (there's more to it)
gitk
    Tk GUI for Git
```

Conflicts

Use add to mark files as resolved.

```
git diff [--base]
git diff --ours
git diff --theirs
git log --merge
gitk --merge
```

Tracking Files

```
git add files
git mv old new
git rm files
git rm --cached files
    (stop tracking but keep files in working dir)
```

Structure Overview

Visual Git Cheat Sheet: https://ndpsoftware.com/git-cheatsheet.html#loc=remote_repo;

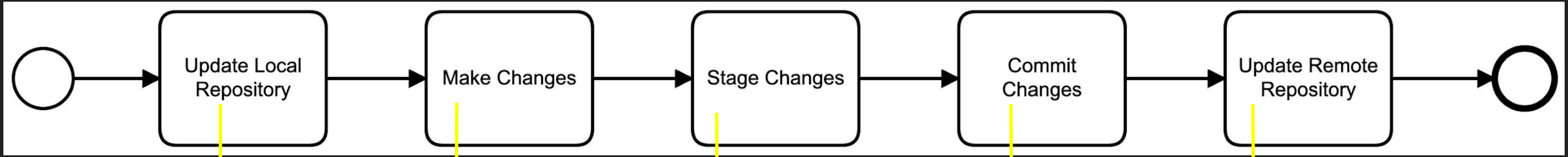
Remotes in Git

A remote repository is a repository stored somewhere else.

Most programmers use hosting services like:

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

Collaborative Workflow



```
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
```

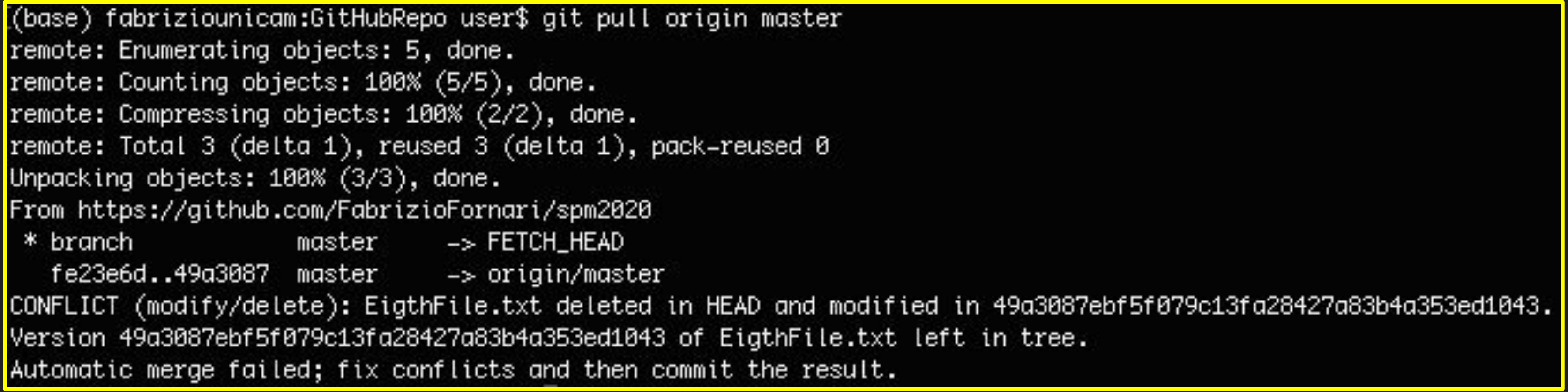
The Dark Side of Collaboration

When you are collaborating on a shared repository you may end up changing the same files other collaborators are working on. Version control helps us to manage these conflicts by giving us tools to resolve overlapping changes.

Note: this can also happen if you are the only one working on a repo but, for instance, on different devices (a laptop and a pc)

Solve the Conflict

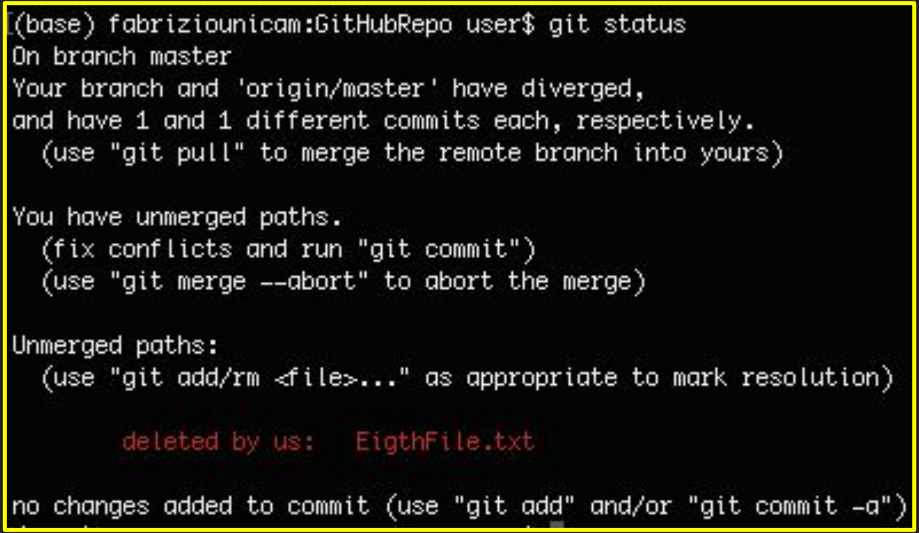
What we have to do is pull the changes from GitHub, merge them into the copy we're currently working in, and then push that.



```
(base) fabriziounicam:GitHubRepo user$ git pull origin master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 3 (delta 1), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/FabrizioFornari/spm2020
 * branch            master     -> FETCH_HEAD
   fe23e6d..49a3087  master     -> origin/master
CONFLICT (modify/delete): EigthFile.txt deleted in HEAD and modified in 49a3087ebf5f079c13fa28427a83b4a353ed1043.
Version 49a3087ebf5f079c13fa28427a83b4a353ed1043 of EigthFile.txt left in tree.
Automatic merge failed; fix conflicts and then commit the result.
```

Solve the Conflict

Run git status



```
(base) fabriziounicam:GitHubRepo user$ git status
On branch master
Your branch and 'origin/master' have diverged,
and have 1 and 1 different commits each, respectively.
  (use "git pull" to merge the remote branch into yours)

You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add/rm <file>..." as appropriate to mark resolution)

        deleted by us:   EigthFile.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

Anything that has merge conflicts and hasn’t been resolved is listed as unmerged. Git adds standard conflict-resolution markers to the files that have conflicts, so you can open them manually and resolve those conflicts.

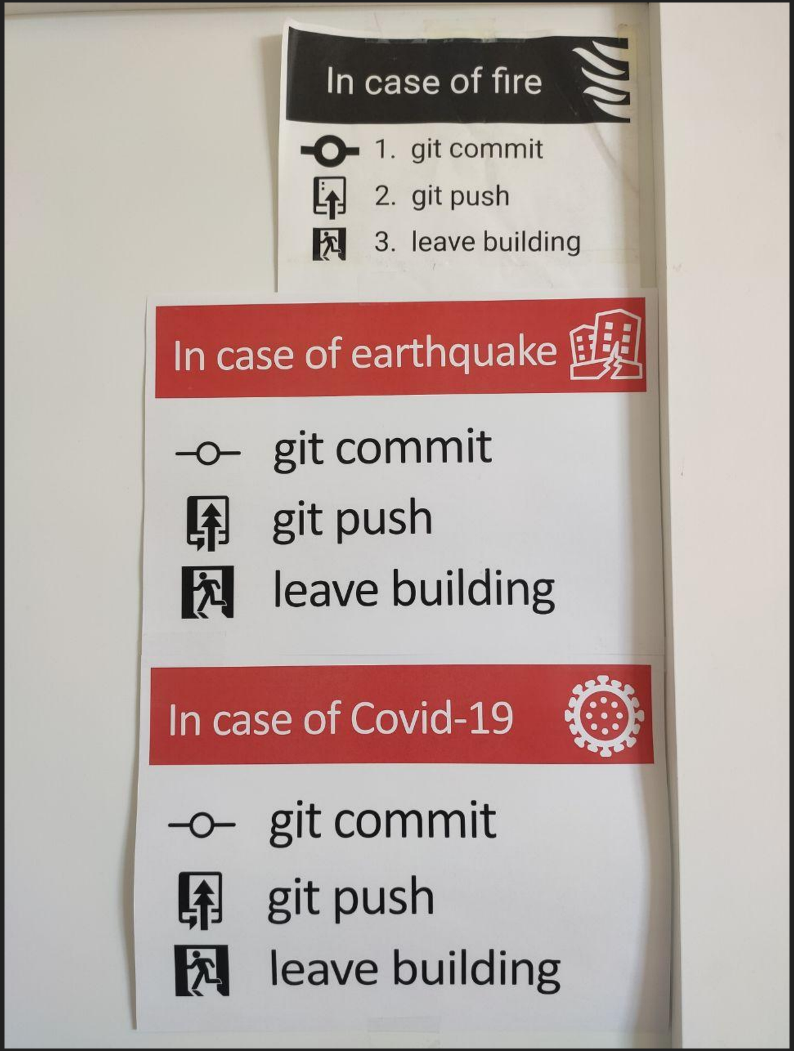
Good Practice

- Pull from upstream more frequently, especially before starting new work
- Use topic branches to segregate work, merging to master when complete
- Make small commits
- Break large files into smaller ones so to reduce the possibility of conflicts

About Conflicts:

- Clarify who is responsible for what areas with your collaborators
- Discuss the order of tasks with your collaborators so that tasks expected to change the same lines won't be worked on simultaneously
- If the conflicts are stylistic churn (e.g. tabs vs. spaces), establish a project convention that is governing and use code style tools (e.g. htmltidy, perltidy, rubocop, etc.) to enforce, if necessary

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

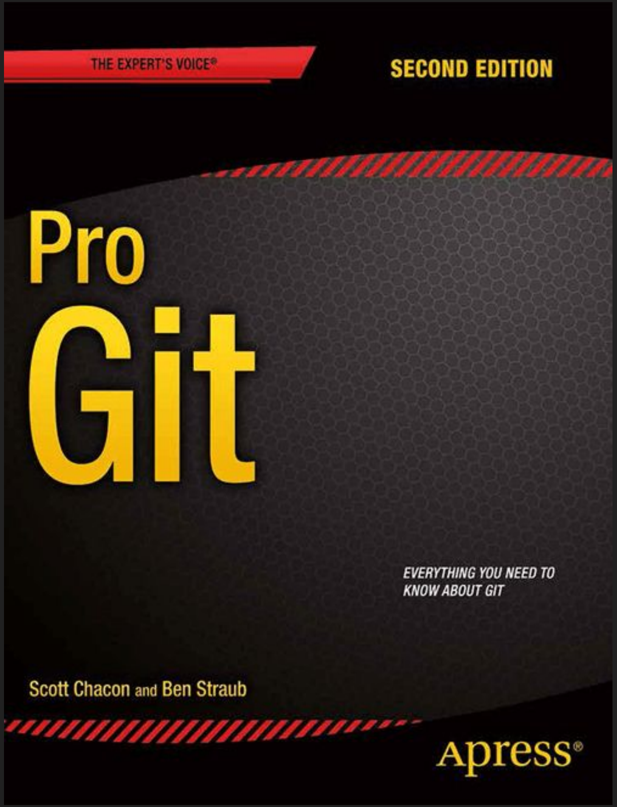
git commit

git push

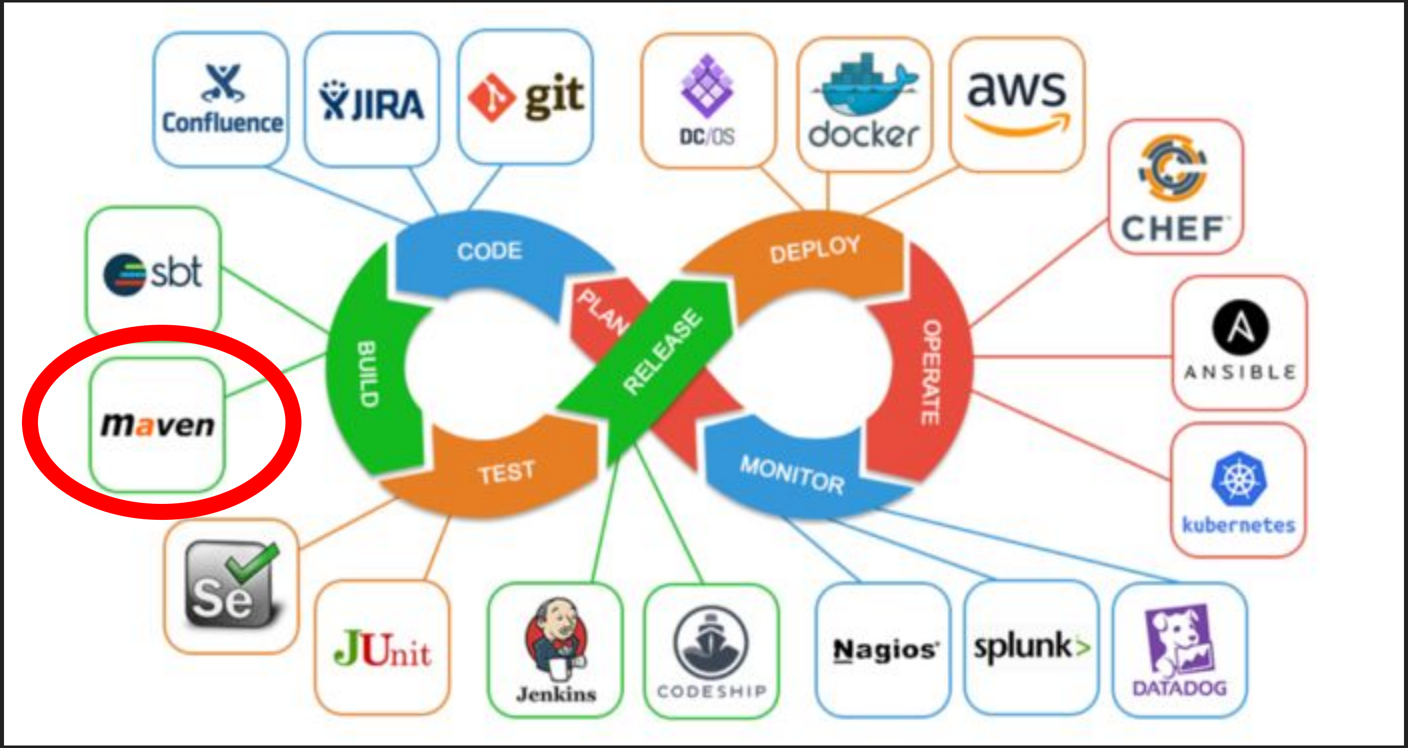
leave building

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®

DevOps


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

Apache Maven

http://maven.apache.org/

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

- Standardized Directory Structure (For example, Maven suggests that all of the Java source code should be placed in the src\main\java folder. This makes it easier to understand and navigate any Maven project.)

- Declarative Dependency Management declare these project dependencies in a separate, external pom.xml file

- Archetypes are predefined project templates that can be used to generate new projects.

Maven

Despite growing competition from other tools, Maven continues to dominate the build tool space.

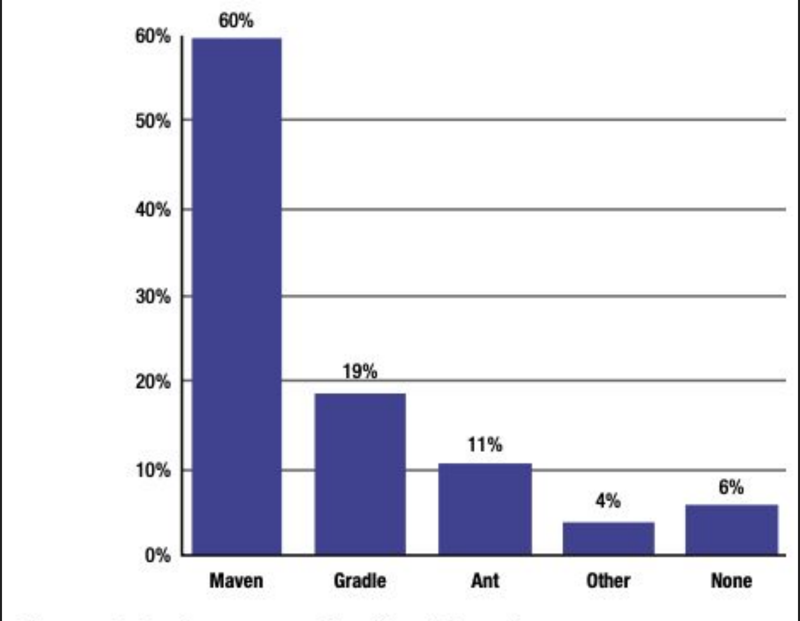
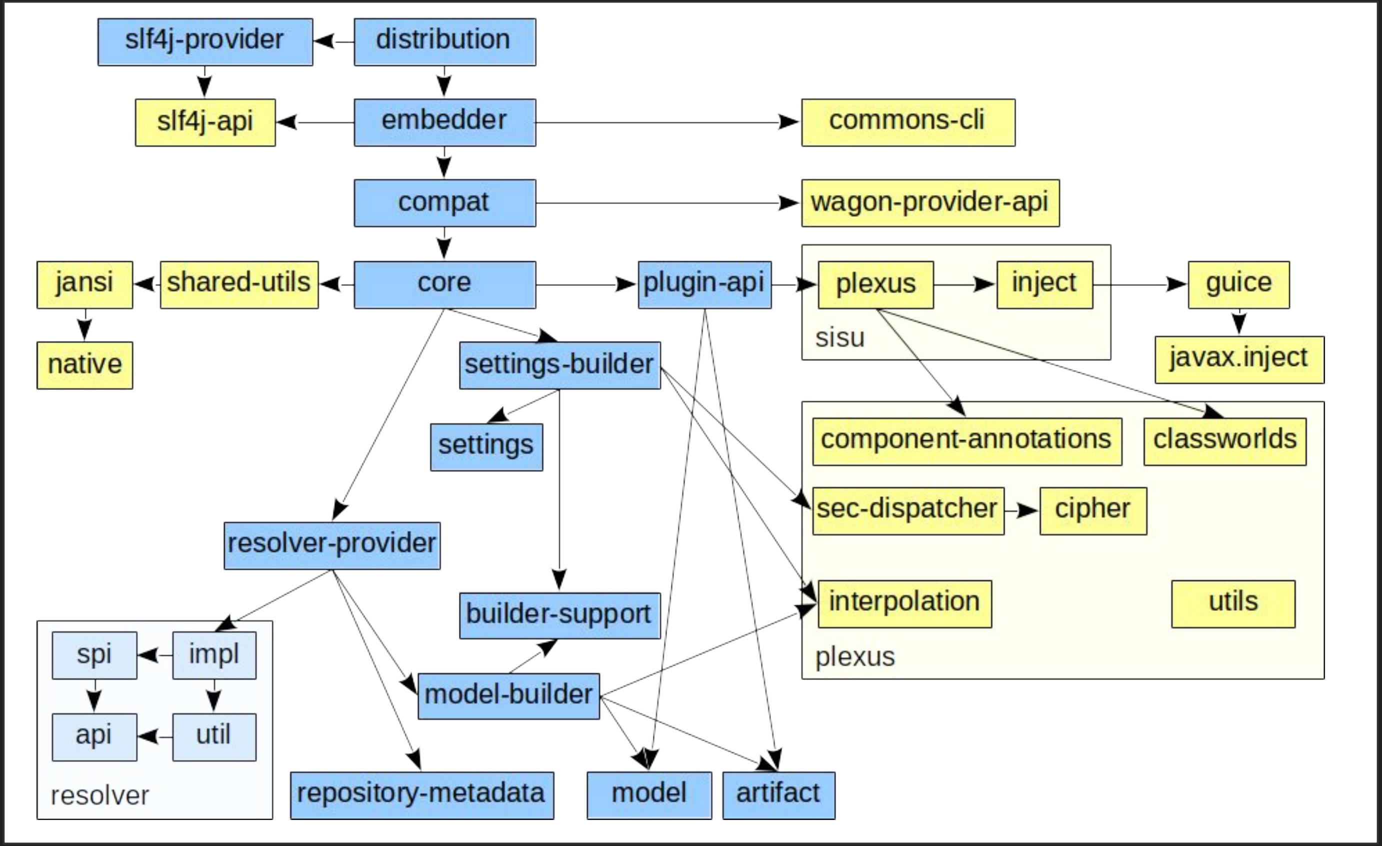


Figure 1-1. *Survey results of build tool usage*

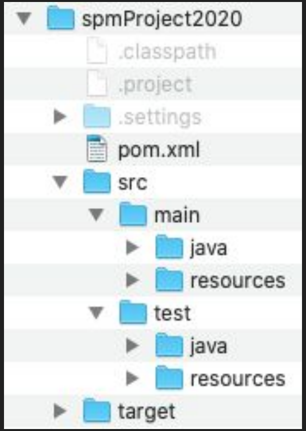
https://snyk.io/blog/jvm-ecosystem-report-2018-tools/

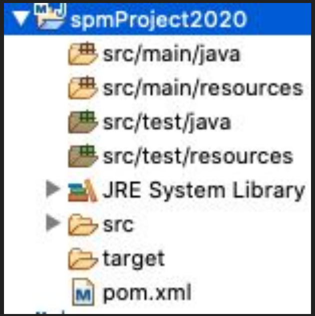
Maven

Maven can be extended by plugins to utilise a number of other development tools for reporting or the build process



Maven - Convention over Configuration





- **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).

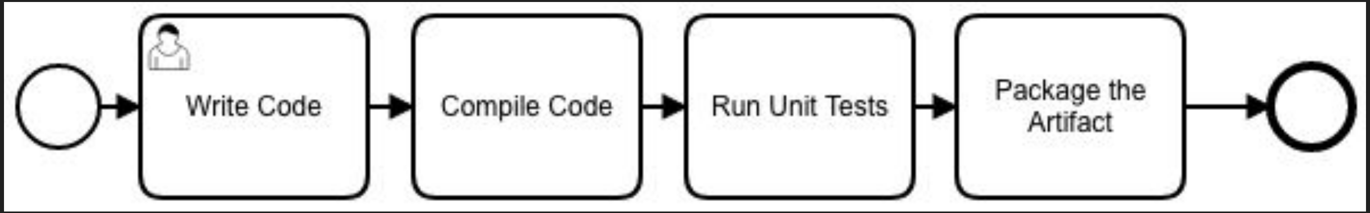
POM also contains the goals and plugins. While executing a task or goal, Maven looks for the POM in the current directory. It reads the POM, gets the needed configuration information, and then executes the goal.

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

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

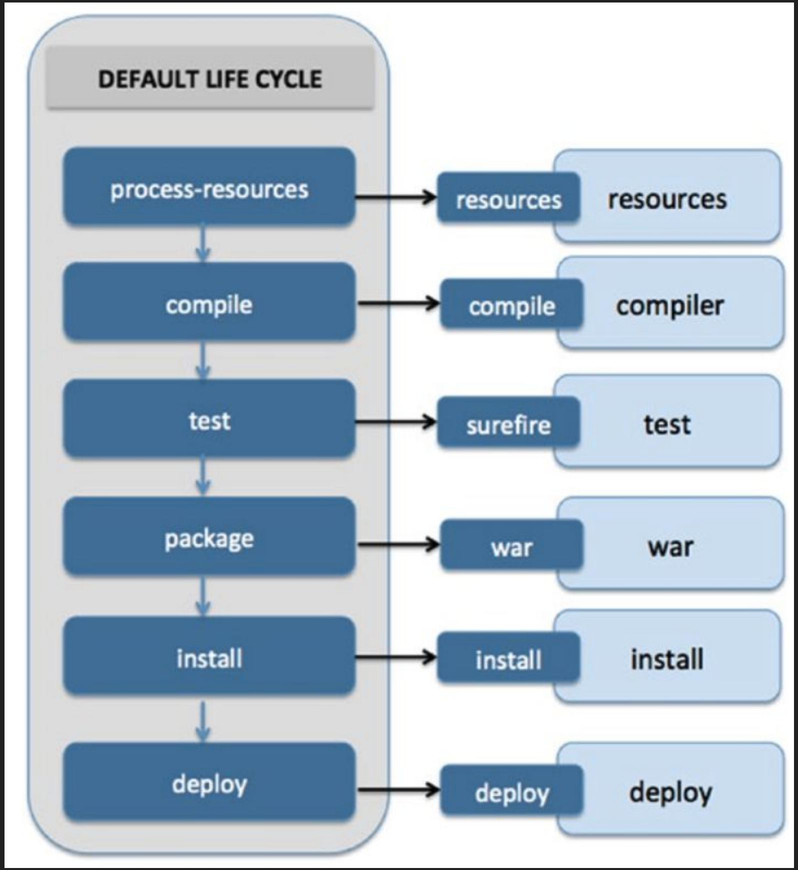
Maven Lifecycle

Build processes generating artifacts such as JAR or WAR files typically require several steps and tasks to be completed successfully in a well-defined order. Examples of such tasks include compiling source code, running unit tests, and packaging of the artifact. Maven uses the concept of goals to represent such granular tasks.



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

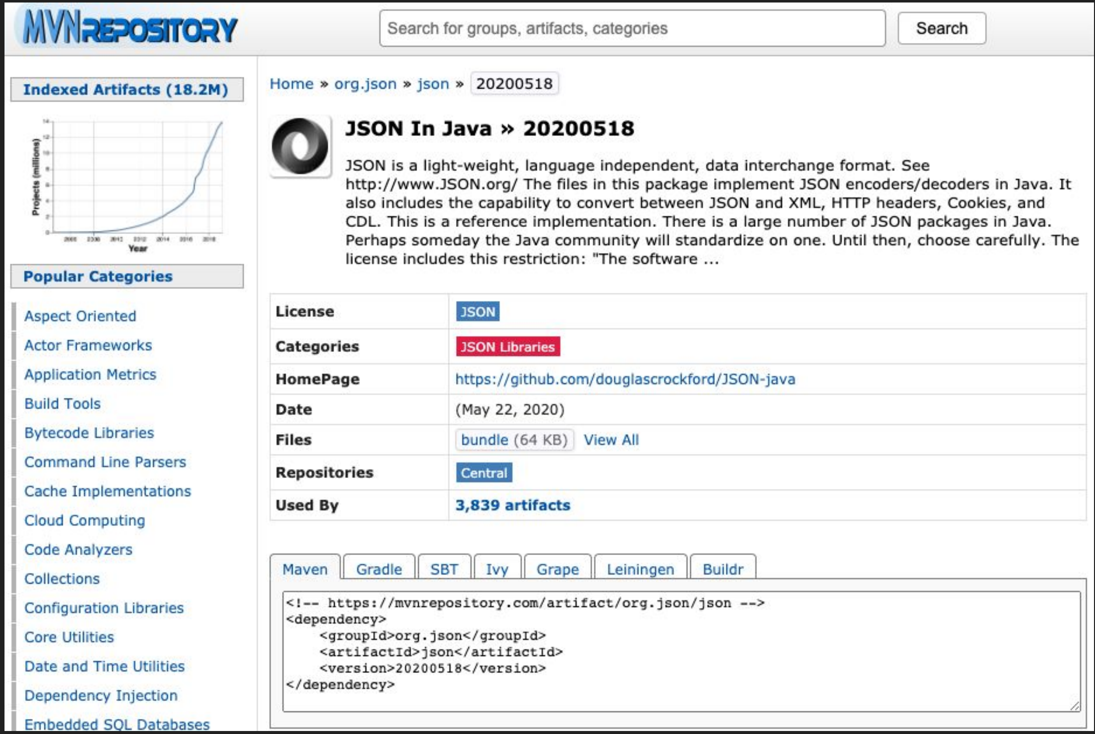
MVNREPOSITORY

https://mvnrepository.com/

Search the library you need
and add it to the POM

I searched for a JSON library

Note: if you don’t know about JSON
https://www.json.org/json-en.html



MVNREPOSITORY | Search for groups, artifacts, categories | Search

Indexed Artifacts (18.2M)

Popular Categories

- Aspect Oriented
- Actor Frameworks
- Application Metrics
- Build Tools
- Bytecode Libraries
- Command Line Parsers
- Cache Implementations
- Cloud Computing
- Code Analyzers
- Collections
- Configuration Libraries
- Core Utilities
- Date and Time Utilities
- Dependency Injection
- Embedded SQL Databases

Home » org.json » json » 20200518

JSON In Java » 20200518

JSON is a light-weight, language independent, data interchange format. See http://www.JSON.org/ The files in this package implement JSON encoders/decoders in Java. It also includes the capability to convert between JSON and XML, HTTP headers, Cookies, and CDL. This is a reference implementation. There is a large number of JSON packages in Java. Perhaps someday the Java community will standardize on one. Until then, choose carefully. The license includes this restriction: "The software ...

License	JSON
Categories	JSON Libraries
HomePage	https://github.com/douglascrockford/JSON-java
Date	(May 22, 2020)
Files	bundle (64 KB) View All
Repositories	Central
Used By	3,839 artifacts

Maven | Gradle | SBT | Ivy | Grape | Leiningen | Buildr

```xml
<!-- https://mvnrepository.com/artifact/org.json/json -->
<dependency>
    <groupId>org.json</groupId>
    <artifactId>json</artifactId>
    <version>20200518</version>
</dependency>
```

Maven - Dependency Management

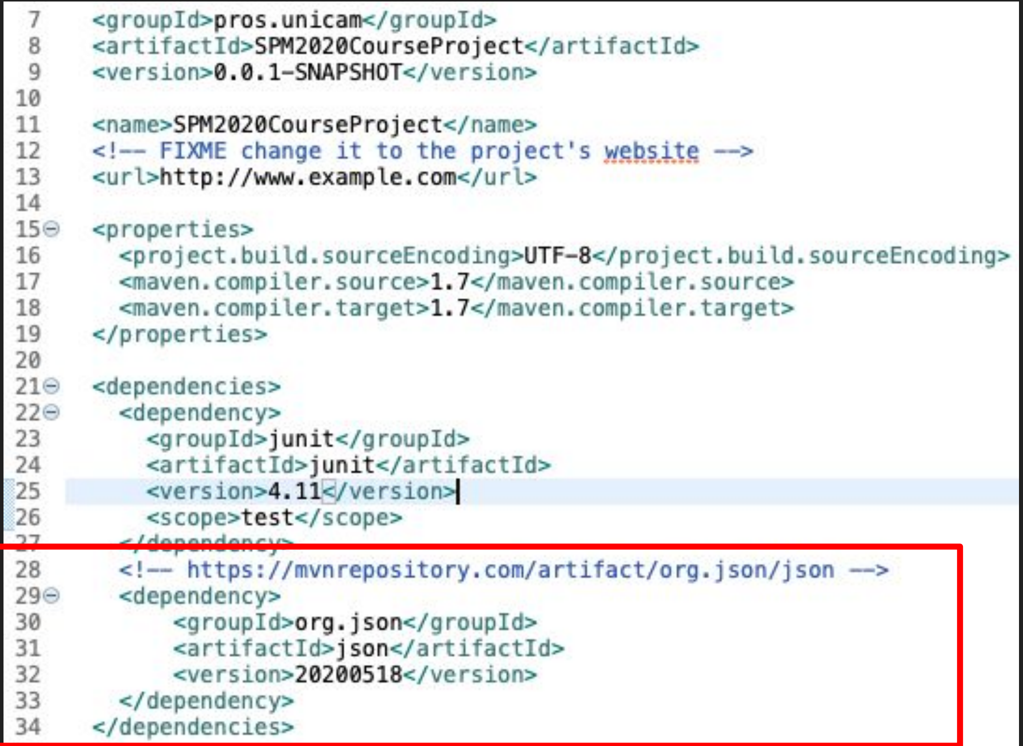
MVNREPOSITORY

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



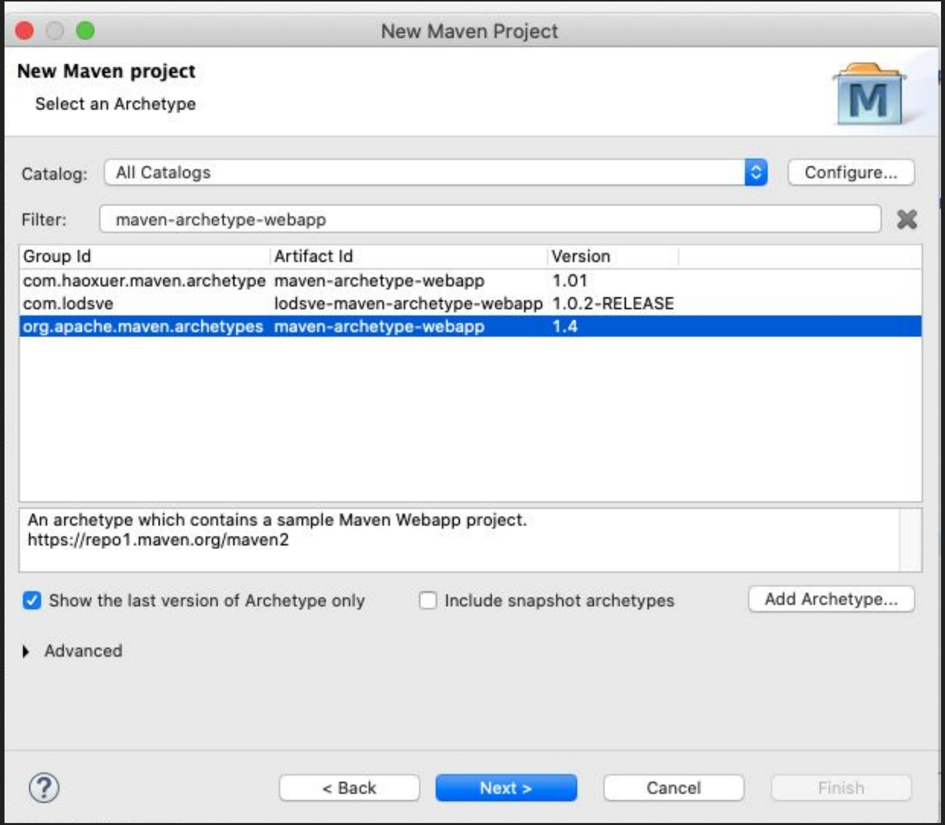
```
<groupId>pros.unicam</groupId>
<artifactId>SPM2020CourseProject</artifactId>
<version>0.0.1-SNAPSHOT</version>

<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



New Maven Project

New Maven project

Select an Archetype

Catalog: All Catalogs

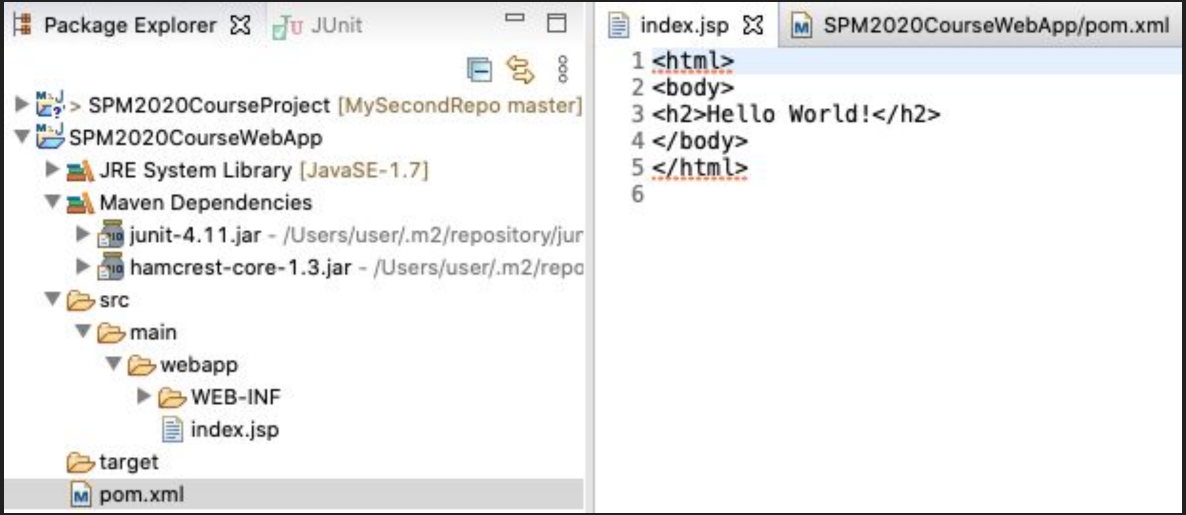
Filter: maven-archetype-webapp

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



```
> 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
```

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

And... now what?

Apache Tomcat

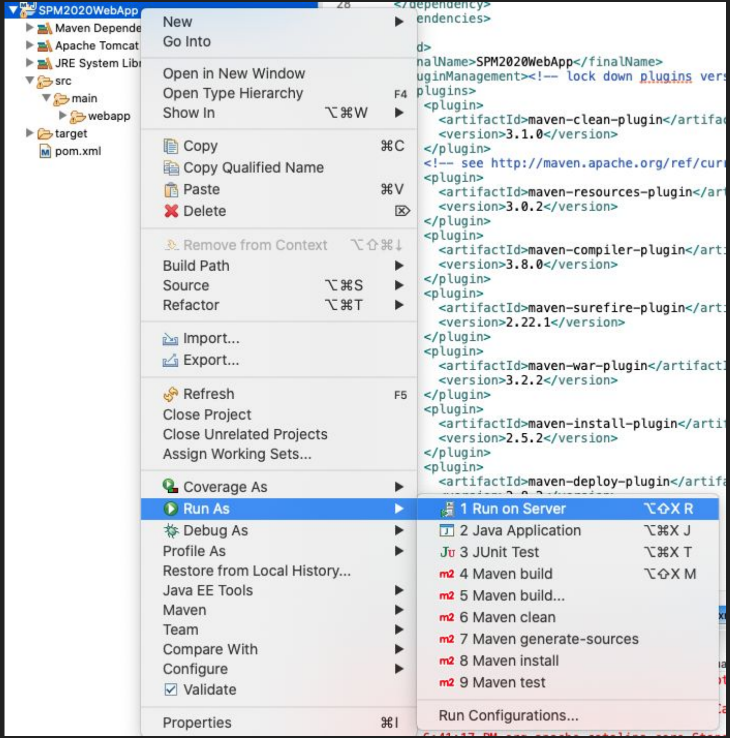
The Apache Tomcat® software is an open source implementation of the Java Servlet, JavaServer Pages, Java Expression Language and Java WebSocket technologies.

http://tomcat.apache.org/

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

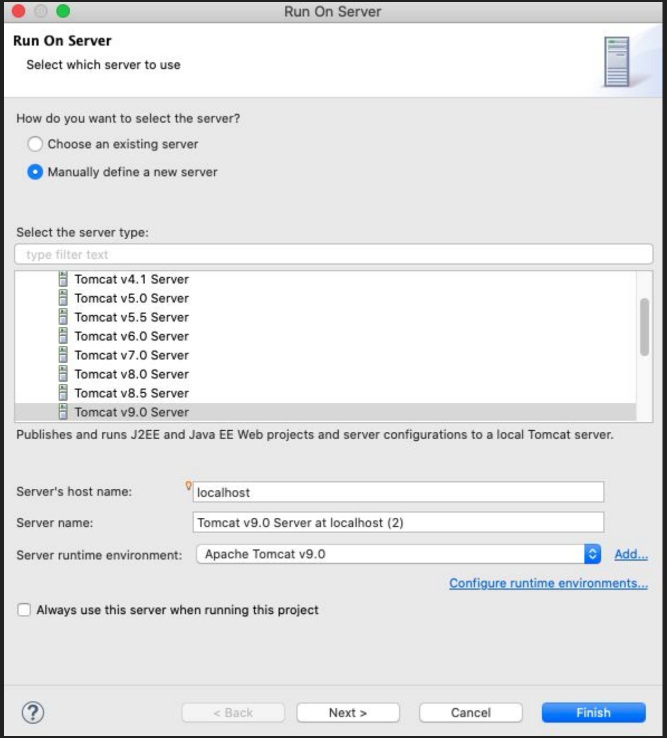
Run Your Application

Run On Server



```
</dependency>
endencies>

d>
nalName>SPM2020WebApp</finalName>
uginManagement><!-- lock down plugins vers
plugins>
  <plugin>
    <artifactId>maven-clean-plugin</artifac
    <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

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

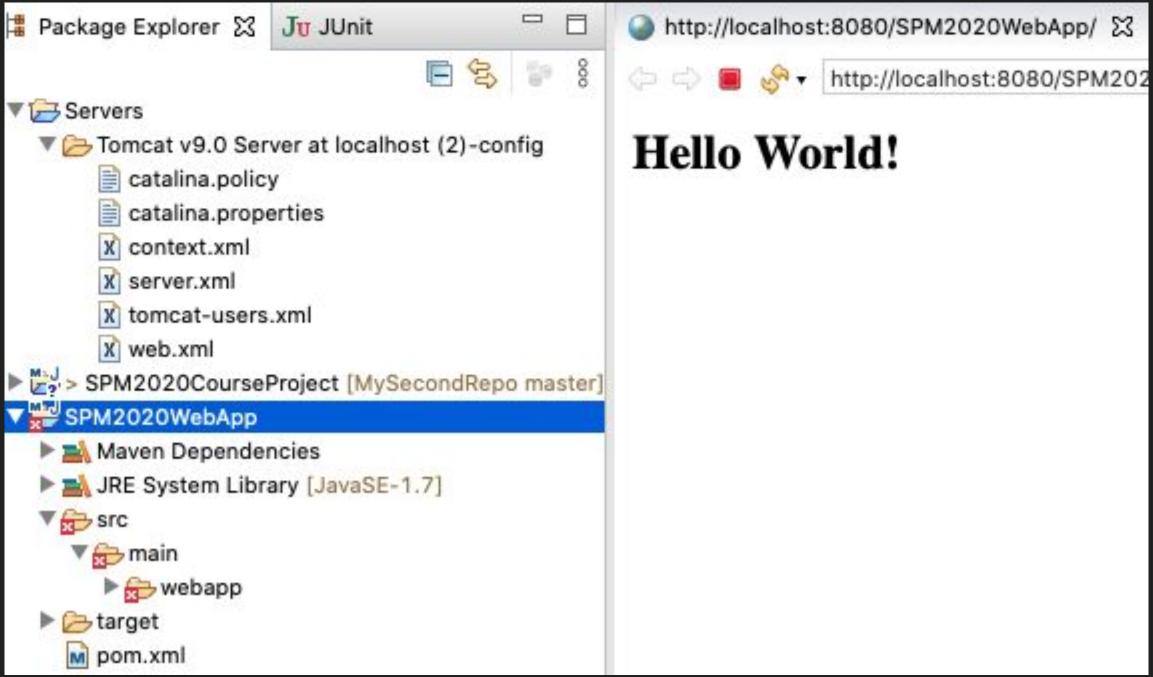
- Tomcat v4.1 Server
- Tomcat v5.0 Server
- Tomcat v5.5 Server
- Tomcat v6.0 Server
- Tomcat v7.0 Server
- Tomcat v8.0 Server
- Tomcat v8.5 Server
- [✓ Tomcat v9.0 Server]

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
http://localhost:8080/SPM2020WebApp/
http://localhost:8080/SPM202

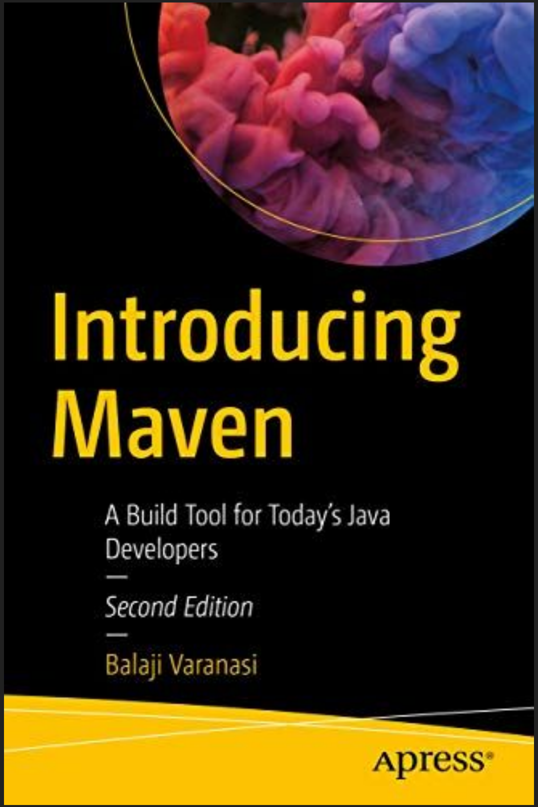
```
▼ Servers
  ▼ Tomcat v9.0 Server at localhost (2)-config
      catalina.policy
      catalina.properties
      context.xml
      server.xml
      tomcat-users.xml
      web.xml
▶ > SPM2020CourseProject [MySecondRepo master]
▼ SPM2020WebApp
  ▶ Maven Dependencies
  ▶ JRE System Library [JavaSE-1.7]
  ▼ src
    ▼ main
      ▶ webapp
  ▶ target
    pom.xml
```

Hello World!

Maven - Additional Material

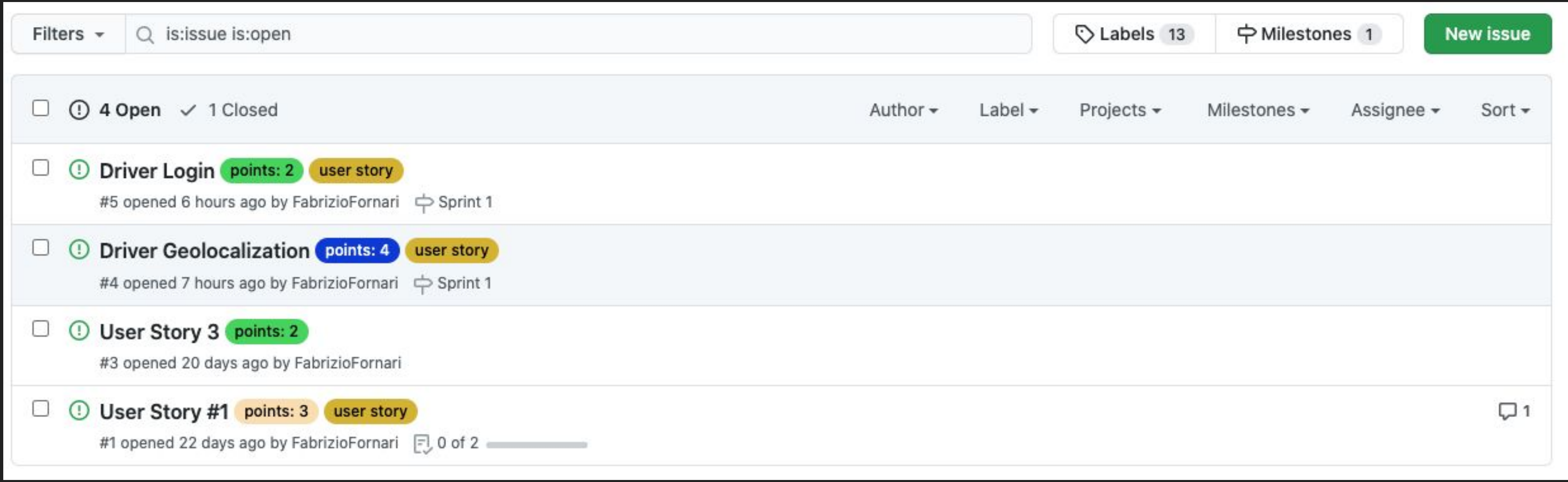
Introducing Maven:
A Build Tool for Today’s Java Developers.

by Balaji Varanasi



Github Project Settings

https://github.com/FabrizioFornari/SPM2020Template



Filters is:issue is:open Labels 13 Milestones 1 New issue

4 Open ✓ 1 Closed Author Label Projects Milestones Assignee Sort

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

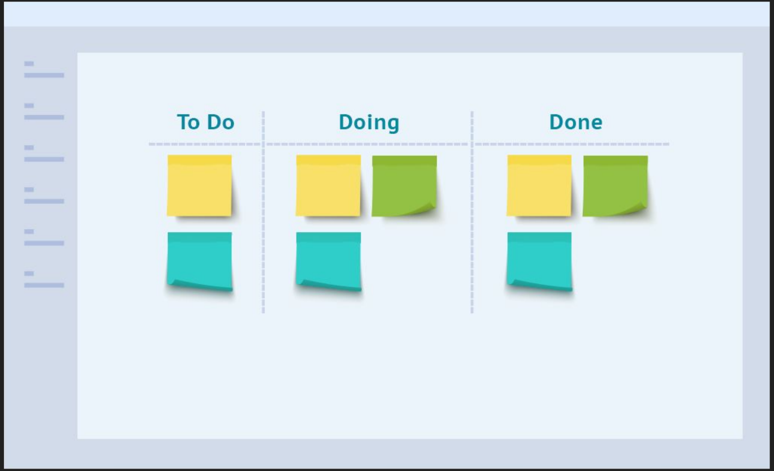
Driver Geolocalization points: 4 user story
#4 opened 7 hours ago by FabrizioFornari Sprint 1

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

Kanban

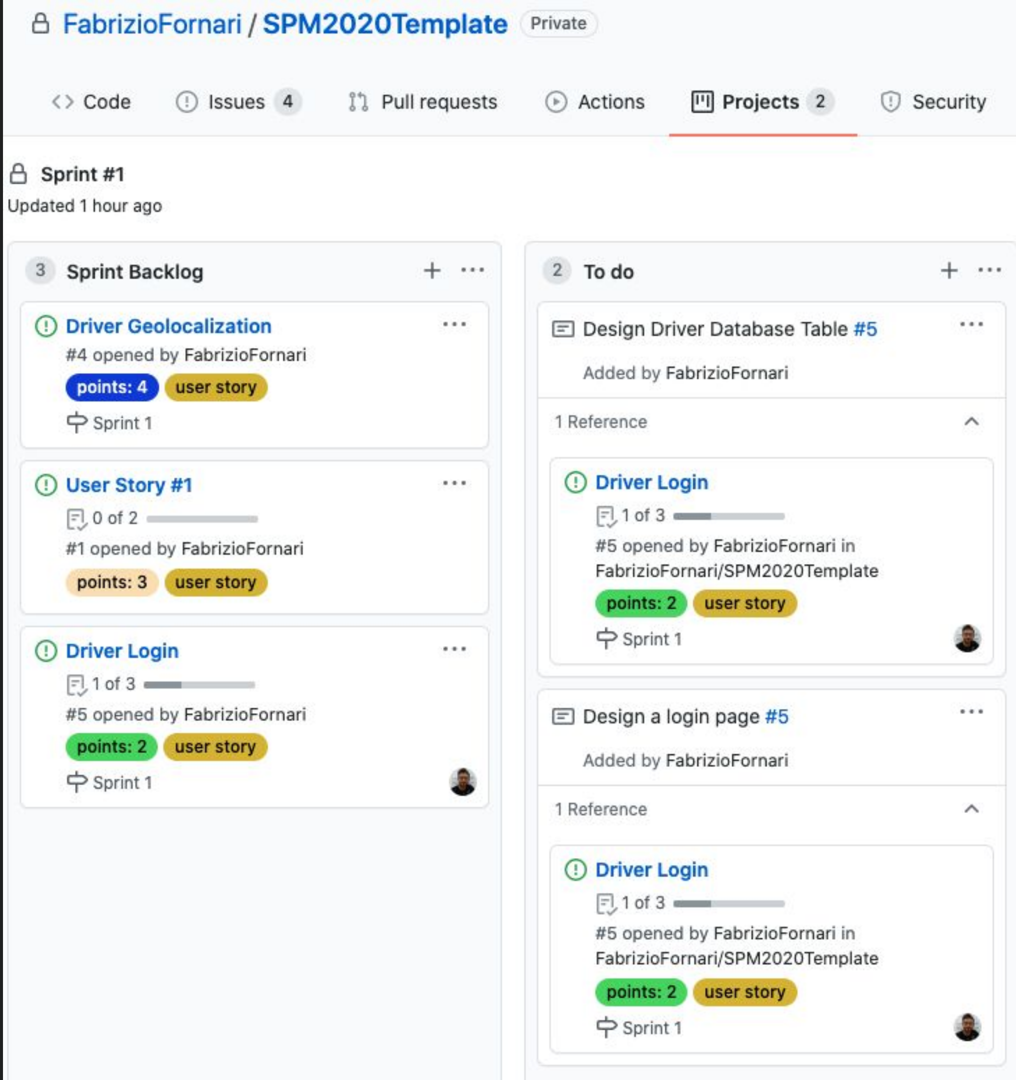
Kanban is a visual system for managing work as it moves through a process. Kanban visualizes both the process (the workflow) and the actual work passing through that process.



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

User Stories vs Tasks
www.mountaingoatsoftware.com



FabrizioFornari / SPM2020Template Private

Code | Issues 4 | Pull requests | Actions | Projects 2 | Security

Sprint #1
Updated 1 hour ago

3 Sprint Backlog

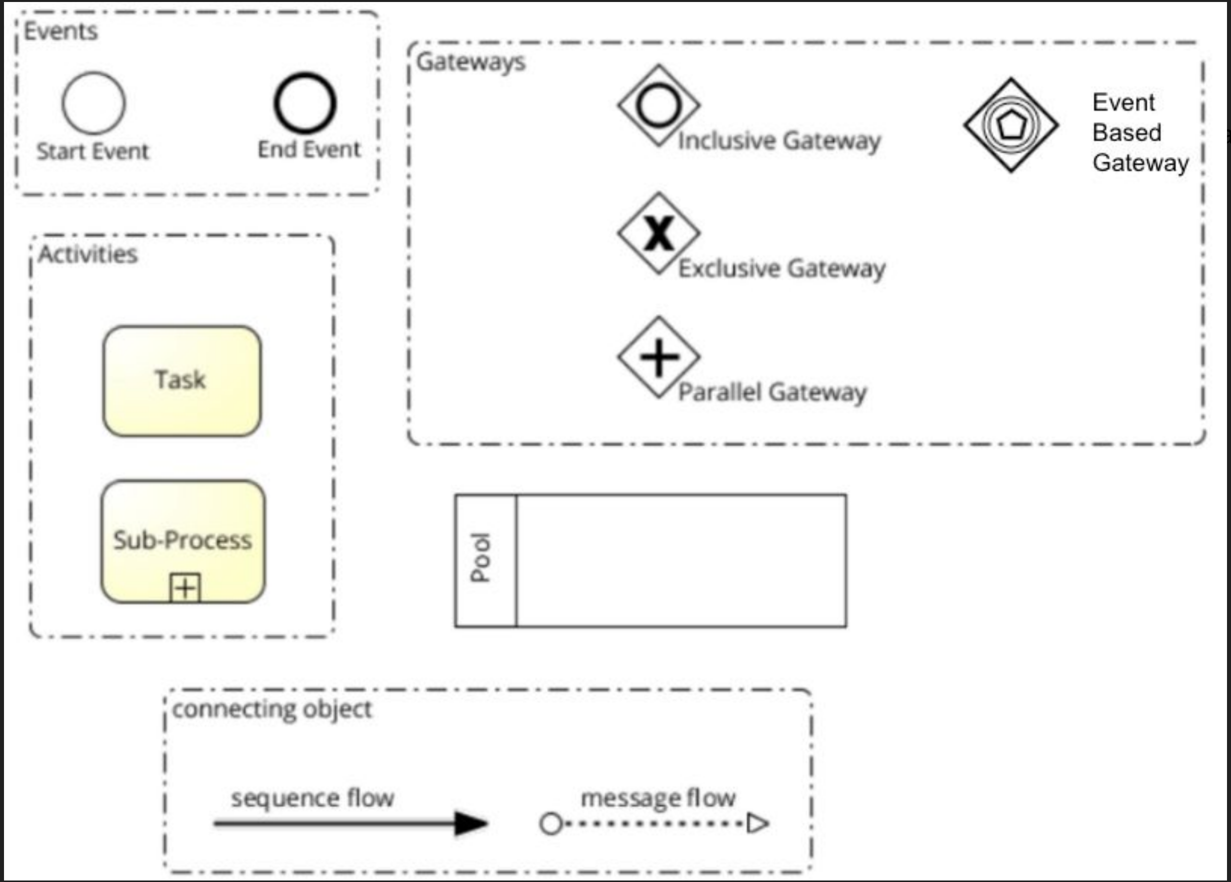
- Driver Geolocalization
  #4 opened by FabrizioFornari
  points: 4 | user story
  Sprint 1
- User Story #1
  0 of 2
  #1 opened by FabrizioFornari
  points: 3 | user story
- Driver Login
  1 of 3
  #5 opened by FabrizioFornari
  points: 2 | user story
  Sprint 1

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

BPMN

Use BPMN to represent the admitted behavior in your application for of each Sprint.



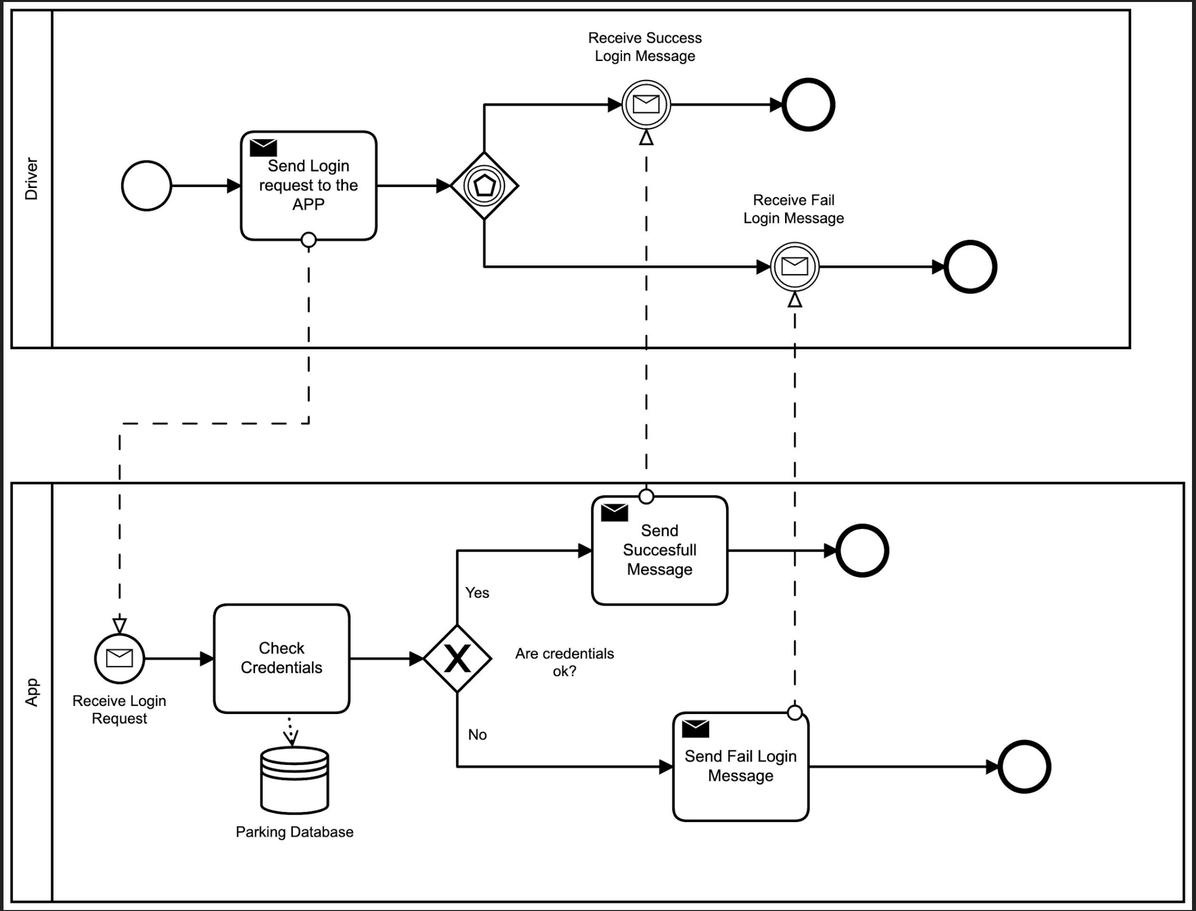
https://www.omg.org/spec/BPMN/2.0/About-BPMN/

BPMN

Reasoning on a Process level can help you in dividing User Stories and Tasks.

You can design BPMN models to describe the user stories your prototype satisfies

CAWEMO - a Collaborative BPMN Modeler
https://cawemo.com/



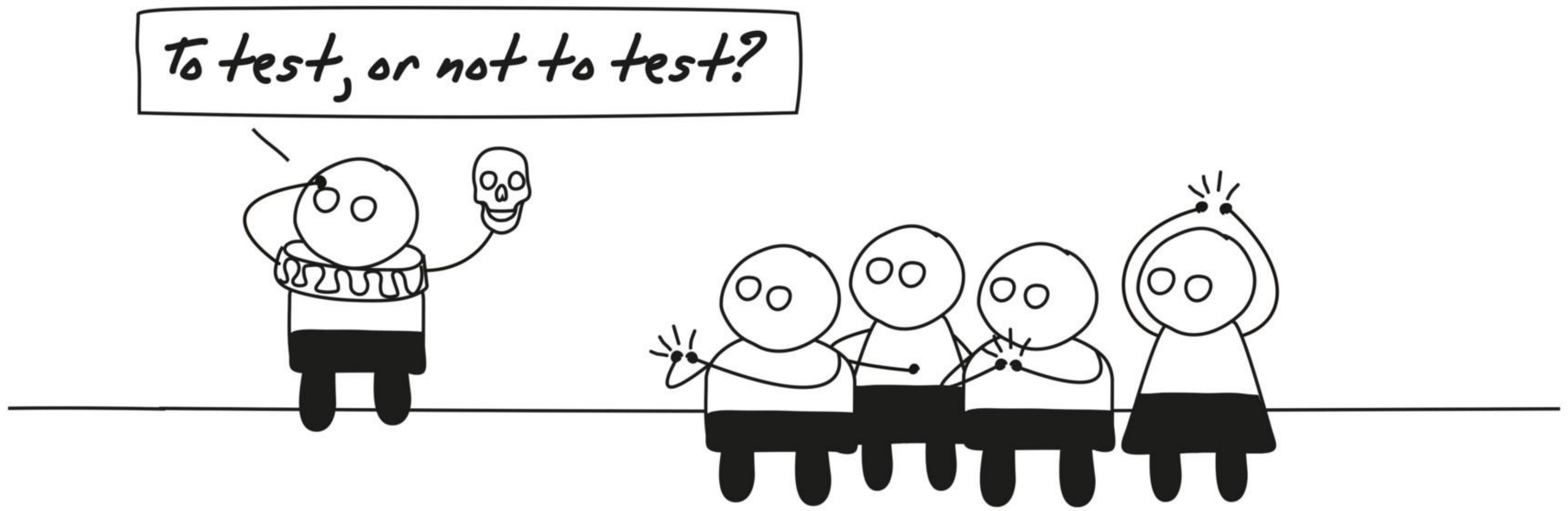
Testing

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 —

AZ QUOTES


To test, or not to test?

Testing

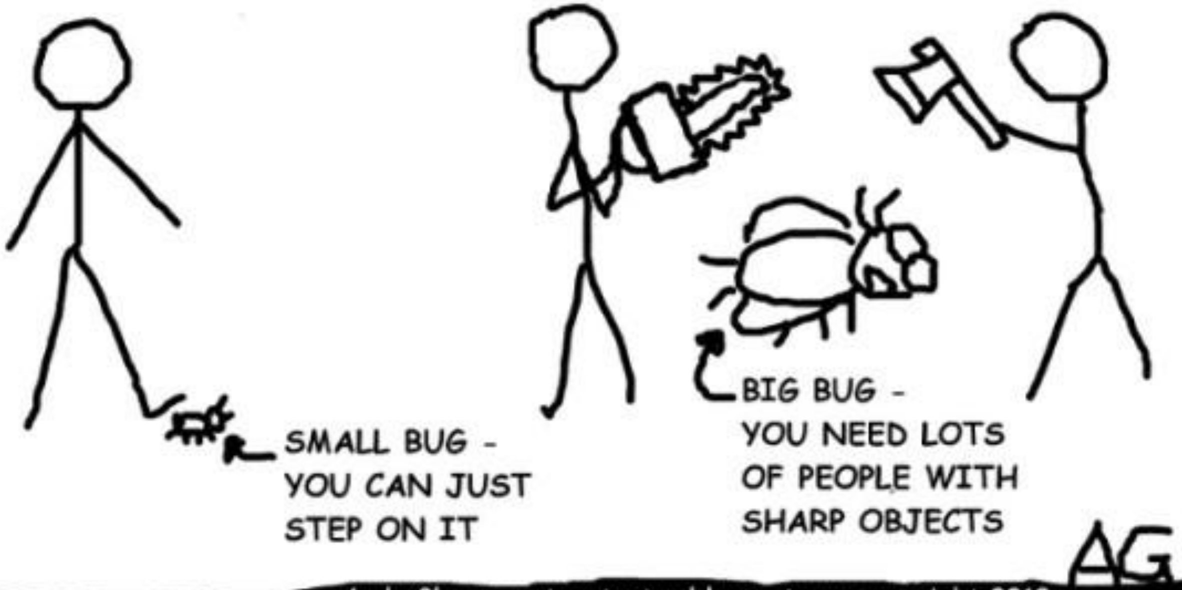
The purpose of testing is to find bugs and errors.

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

Test Sizes

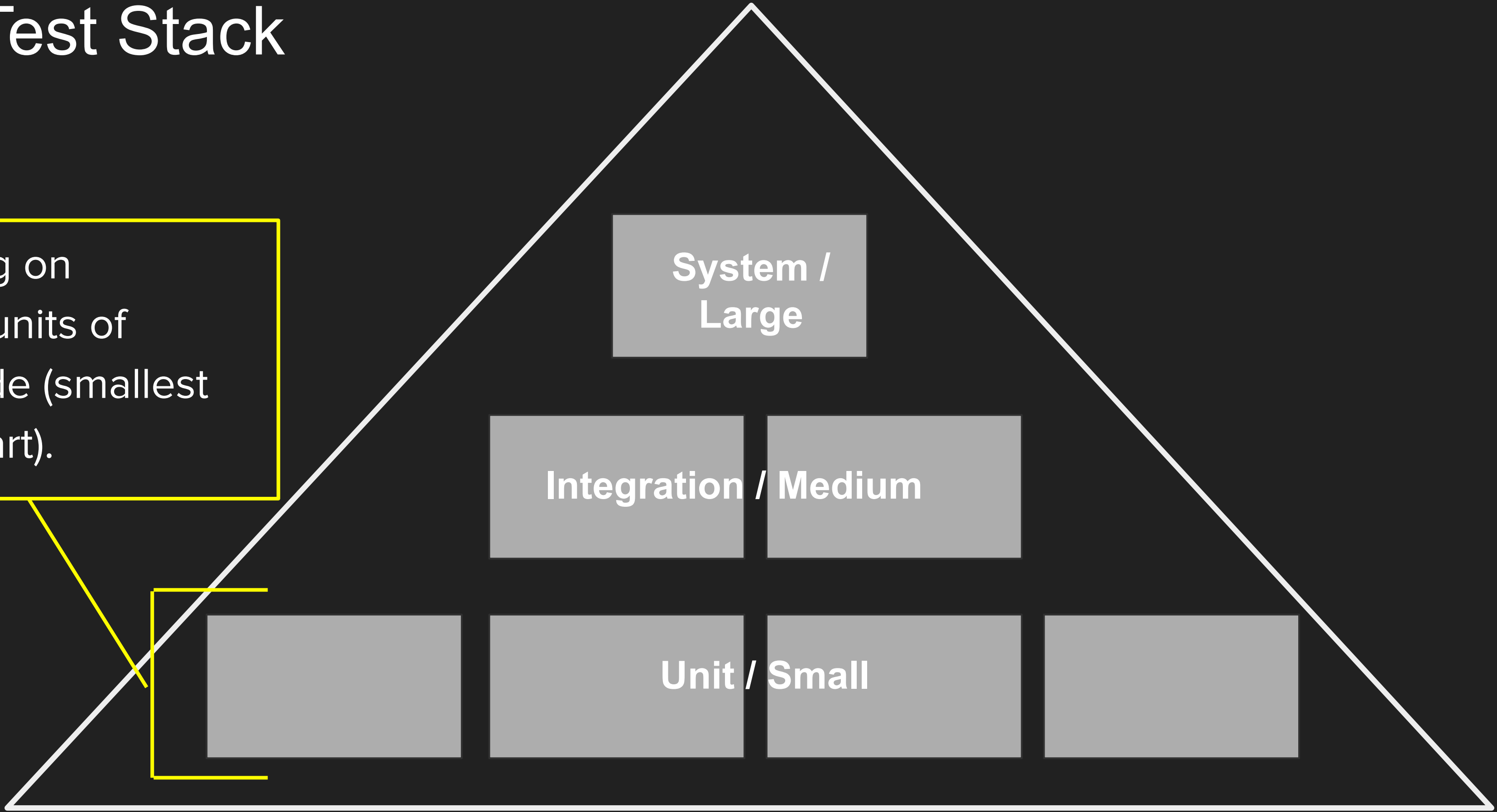
Size & Time →

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+

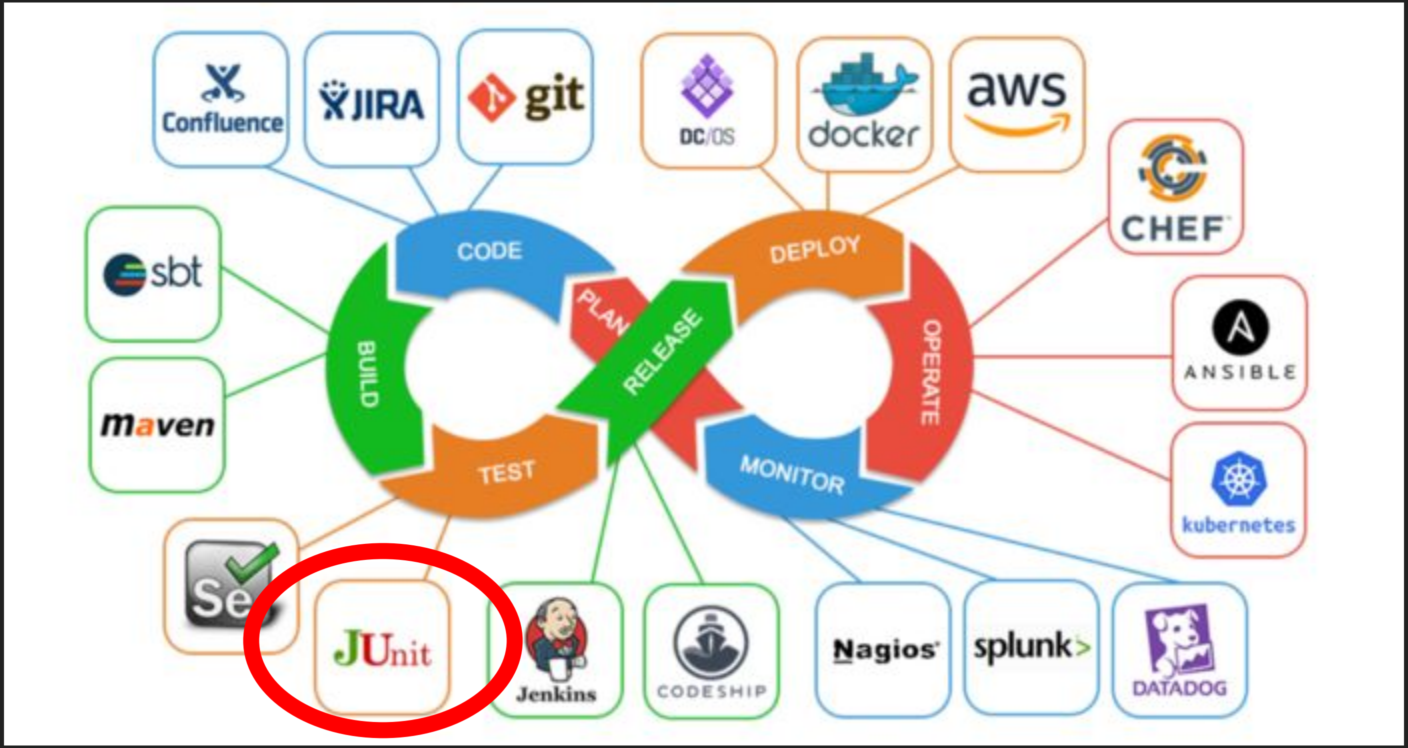
Google Testing Blog 'Test Sizes'

The Test Stack

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



DevOps


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

Unit tests and unit testing

- 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

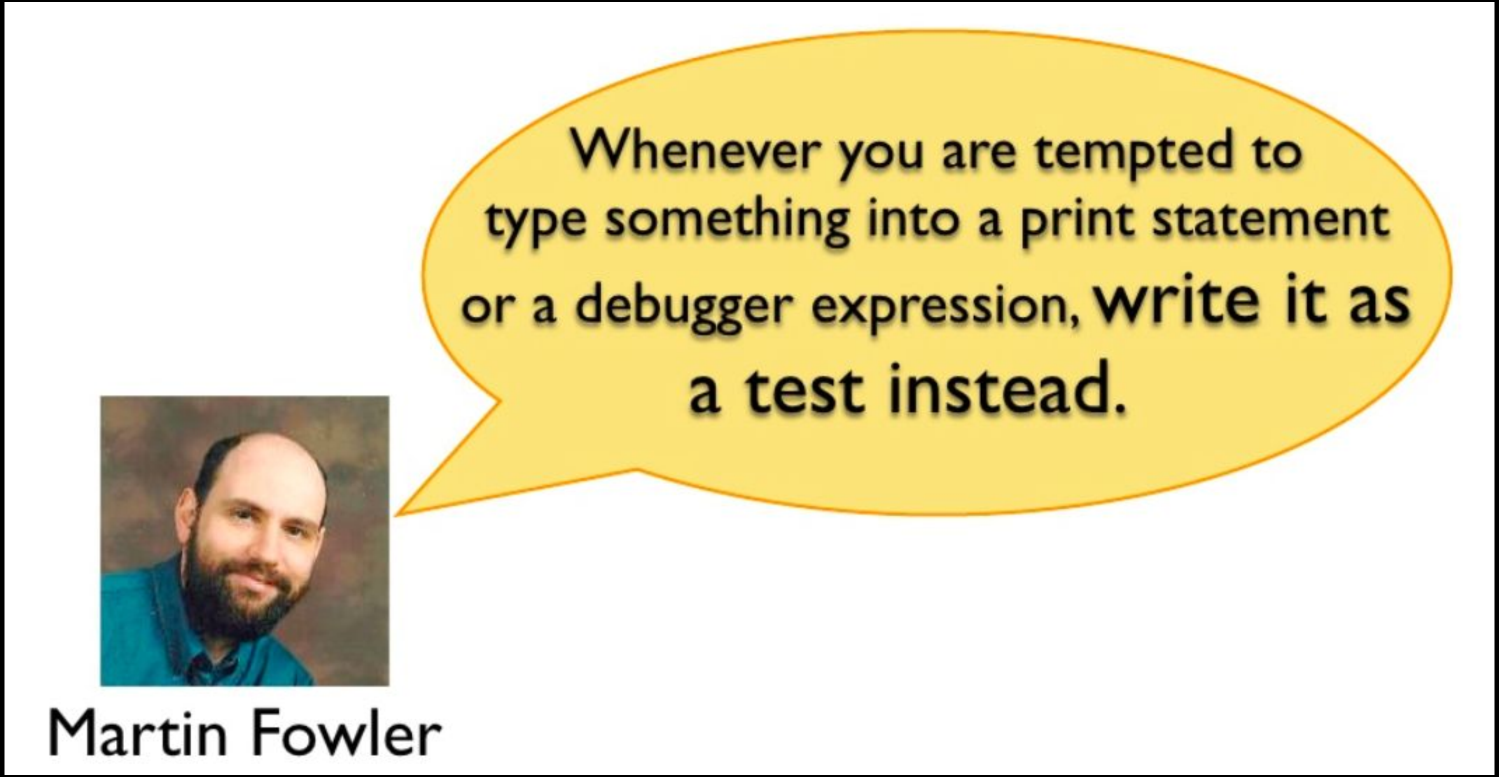
```
package test;

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

Best practices


Whenever you are tempted to type something into a print statement or a debugger expression, write it as a test instead.
Martin Fowler

One of the founding fathers of Extreme Programming

Extreme Programming (XP)

- A type of Agile software development
  - it advocates frequent "releases" in short development cycles
  - introduce checkpoints at which new customer requirements can be adopted

- Other elements of extreme programming include:
  - programming in pairs or doing extensive code review
  - unit testing of all code
  - avoiding programming of features until they are actually needed
  - code simplicity and clarity
  - expecting changes in the customer's requirements as time passes and the problem is better understood
  - frequent communication with the customer and among programmers
- XP uses Test Driven Development (TDD) and refactoring to help uncover the most effective design.
  - refactoring can be safely achieved only with a strong test system, able to check that the whole software product don't break when you add new code, or when you modify existing ones.

Test-Driven Development (TDD)

- Writing test before code to be tested
  - “a little test, a little code, a little test, a little code, ...”
  - Tests are added gradually during implementation – not in large lump afterwards

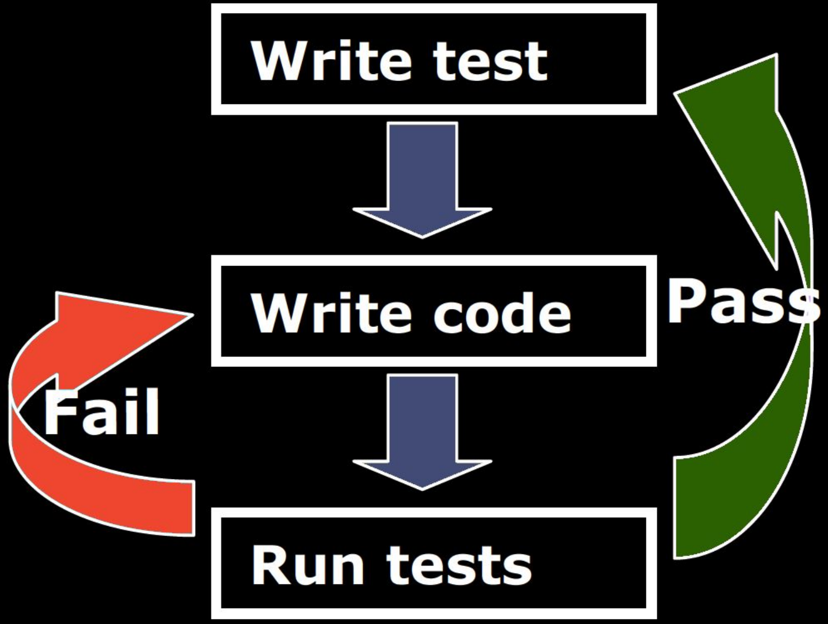
- Process of writing tests drives low-level design and programming
  - Tests specify what code should do
  - Tests validate that code does what it should

- Actually, a design and coding practice
- One of the core practices of Extreme Programming
  - Developers have been applying TDD for several decades

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



and for anything else...

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/

Types of Testing

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

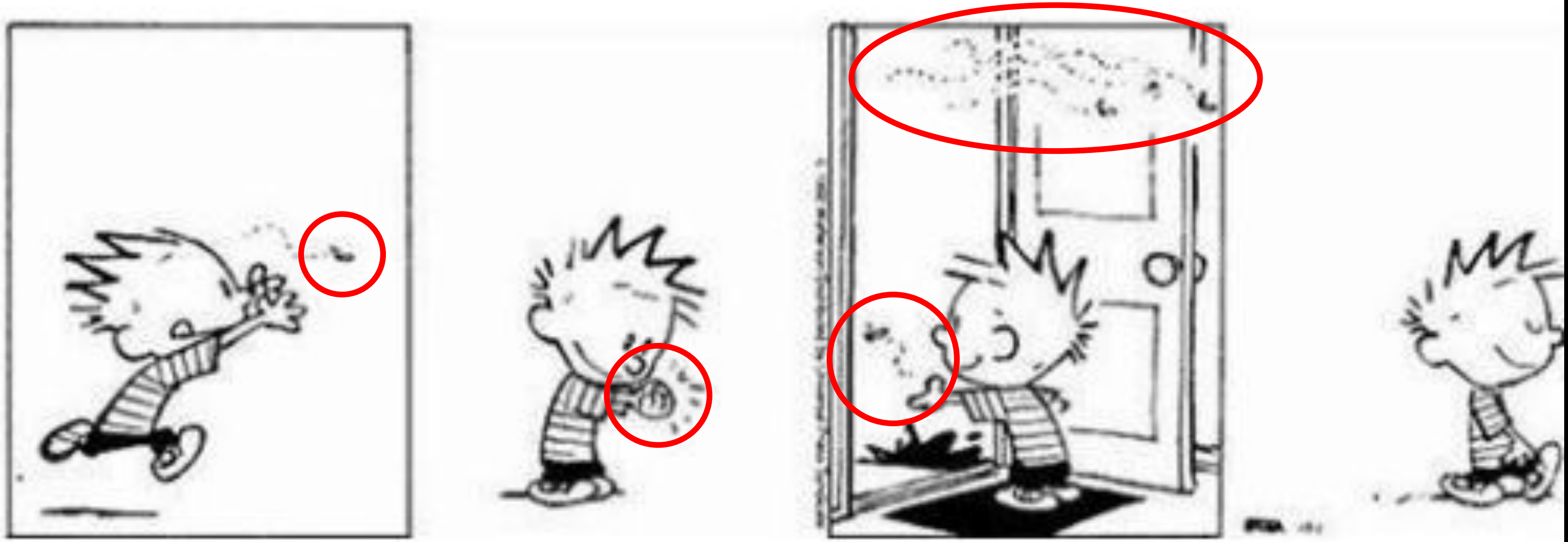
https://www.softwaretestinghelp.com/types-of-software-testing/

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 that too even in a single bug fix.

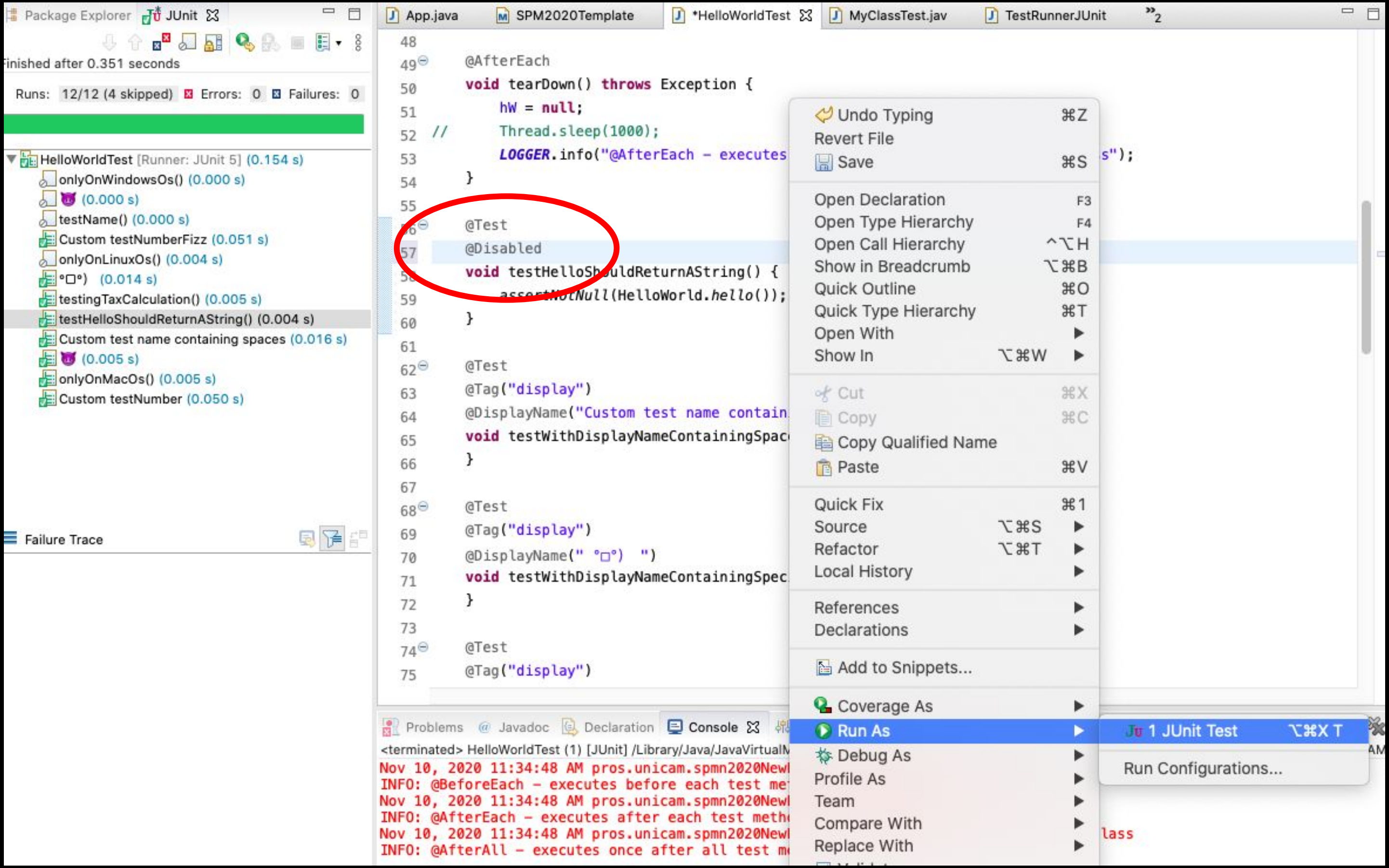
Manual Testing

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



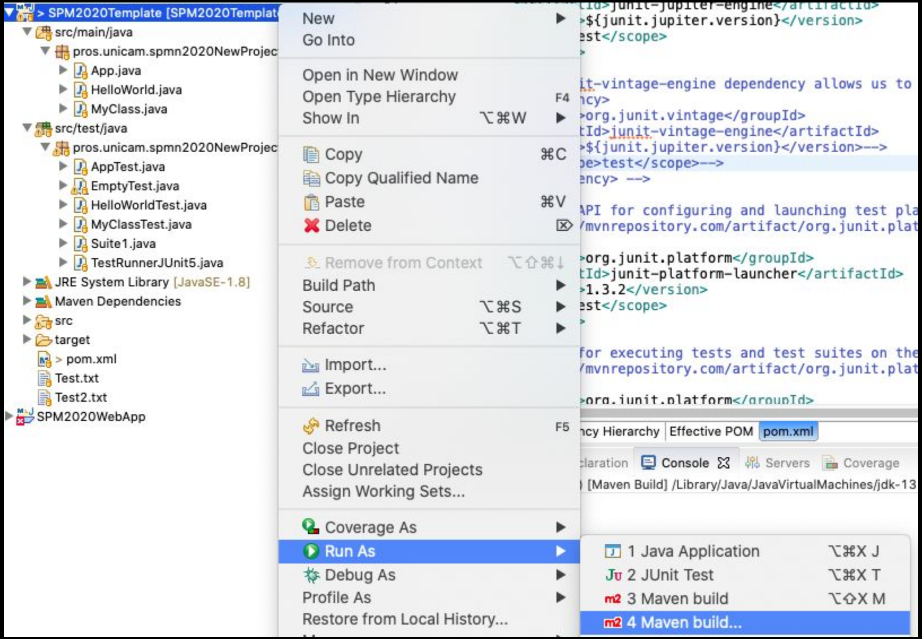
```
inished after 0.351 seconds

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)
```

```java
    @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 m
```

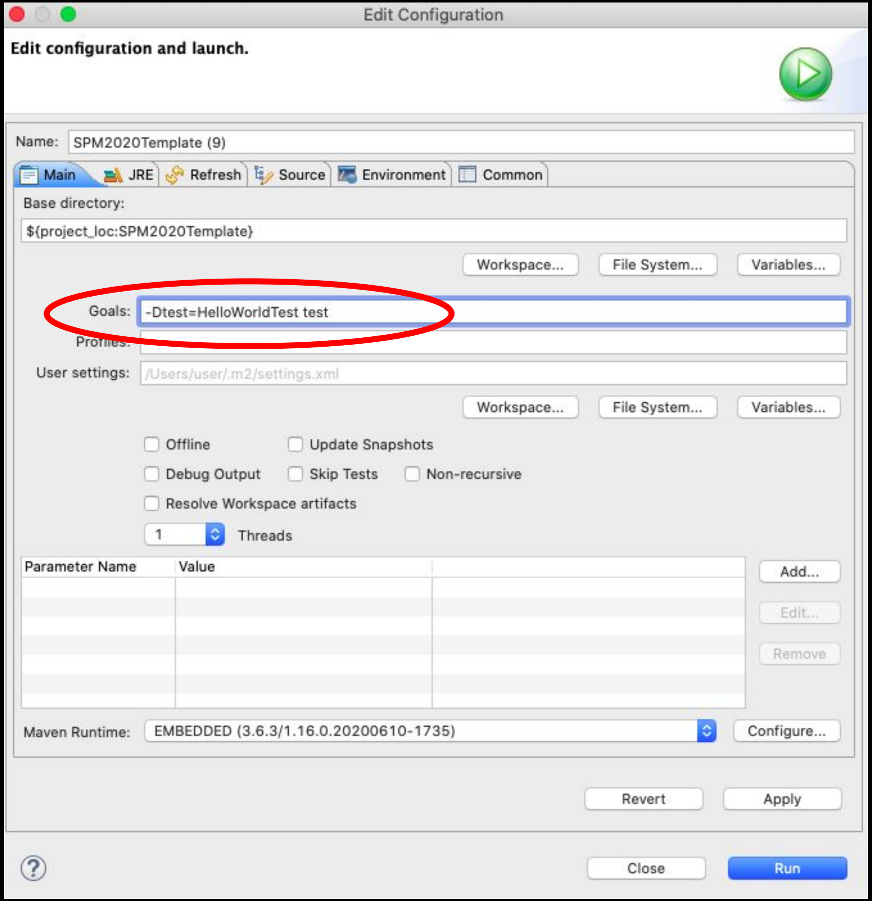
Manual Testing with Maven

- 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
```



Name: SPM2020Template (9)

Base directory: ${project_loc:SPM2020Template}

Goals: -Dtest=HelloWorldTest test

Profiles:

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

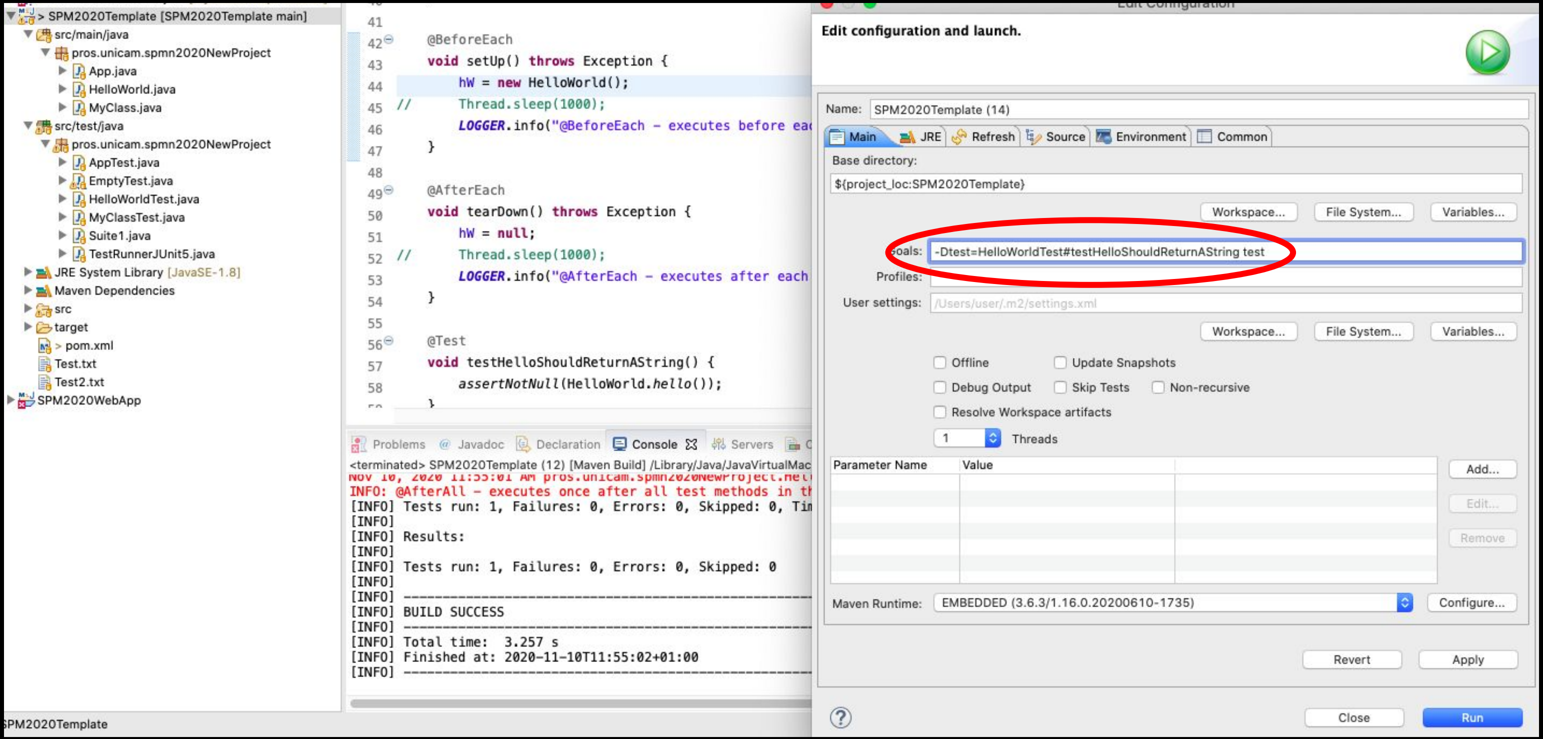
☐ Offline ☐ Update Snapshots
☐ Debug Output ☐ Skip Tests ☐ Non-recursive
☐ Resolve Workspace artifacts
1 Threads

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



```
> SPM2020Template [SPM2020Template main]
  src/main/java
    pros.unicam.spmn2020NewProject
      App.java
      HelloWorld.java
      MyClass.java
  src/test/java
    pros.unicam.spmn2020NewProject
      AppTest.java
      EmptyTest.java
      HelloWorldTest.java
      MyClassTest.java
      Suite1.java
      TestRunnerJUnit5.java
  JRE System Library [JavaSE-1.8]
  Maven Dependencies
  src
  target
  > pom.xml
  Test.txt
  Test2.txt
SPM2020WebApp
```

```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)

Base directory: ${project_loc:SPM2020Template}

Goals: -Dtest=HelloWorldTest#testHelloShouldReturnAString 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)

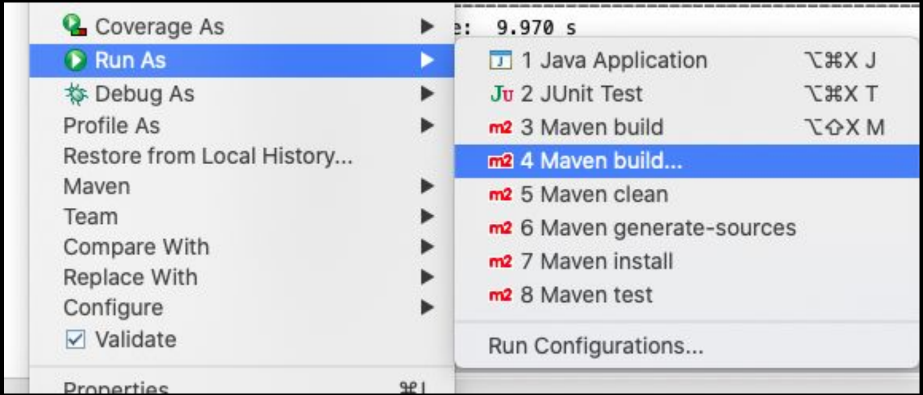
SPM2020Template

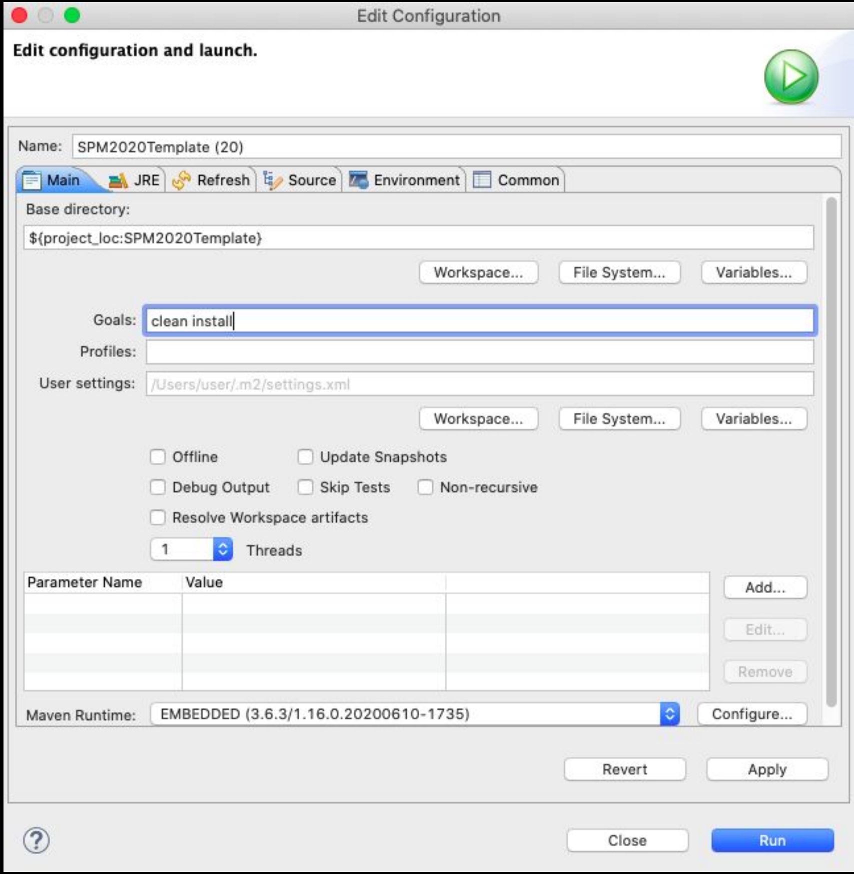
Automated testing

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




Edit Configuration

Edit configuration and launch.

Name: SPM2020Template (20)

Main | JRE | Refresh | Source | Environment | Common

Base directory: ${project_loc:SPM2020Template}

Goals: clean install

Profiles:

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

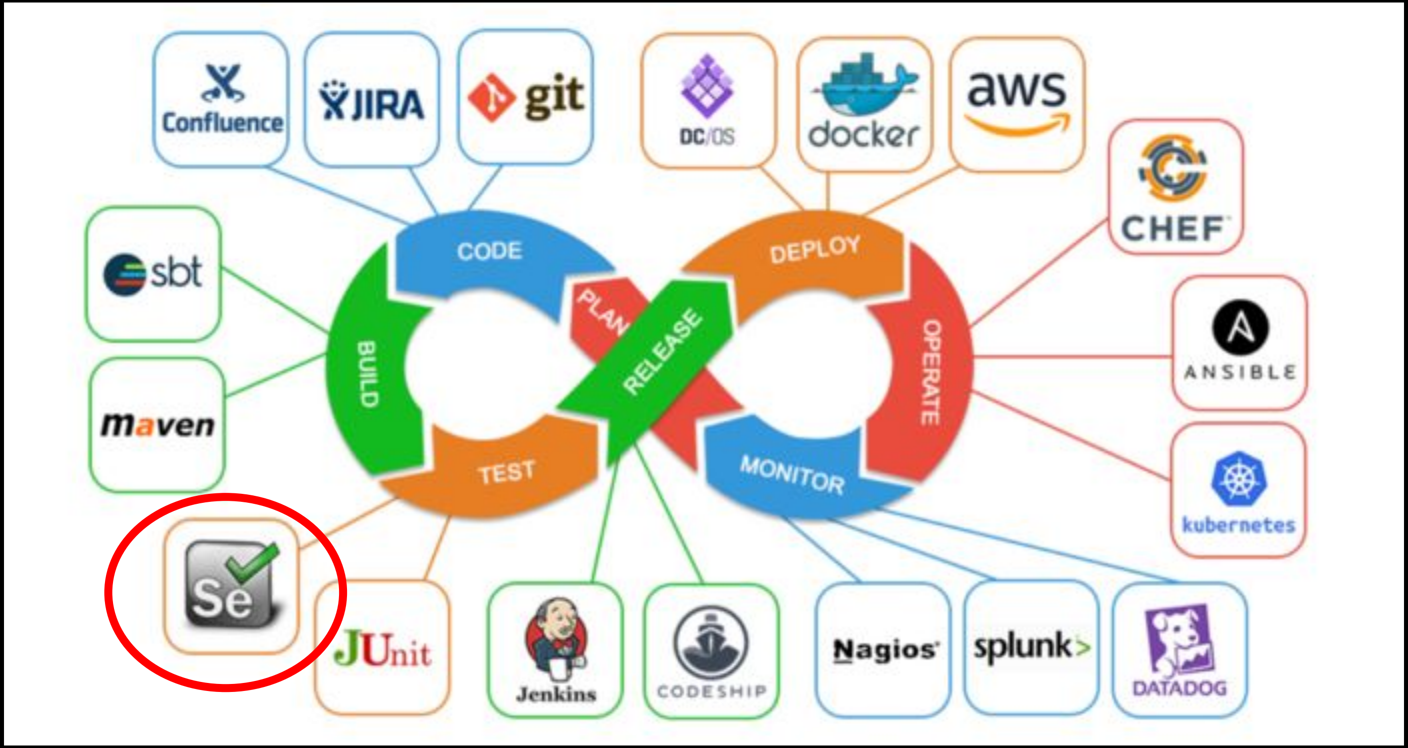
Offline ☐ Update Snapshots ☐
Debug Output ☐ Skip Tests ☐ Non-recursive ☐
Resolve Workspace artifacts ☐

Threads: 1

Parameter Name	Value

Maven Runtime: EMBEDDED (3.6.3/1.16.0.20200610-1735)

DevOps


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

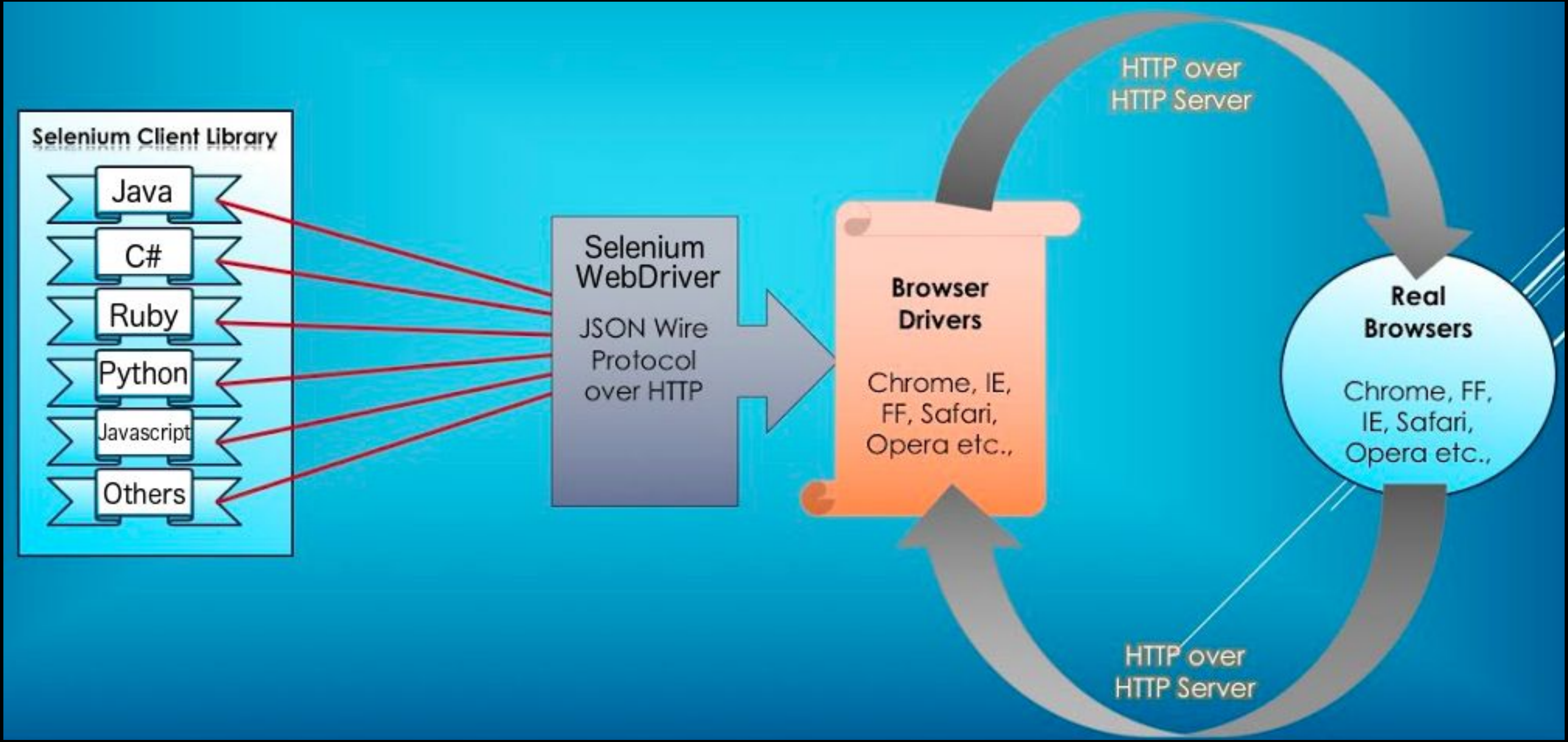
Which is the idea?

To remotely control browsers so that we can do things like write automated tests for the content they run or tests for the browser UI itself.

Should I write a test in a different way per each browser that is out there? No, to this end, a group of people from several organizations is working on the WebDriver Specification.

https://w3c.github.io/webdriver/

Selenium Architecture



How to check the status of HTTP Request?

We can use Rest Assured

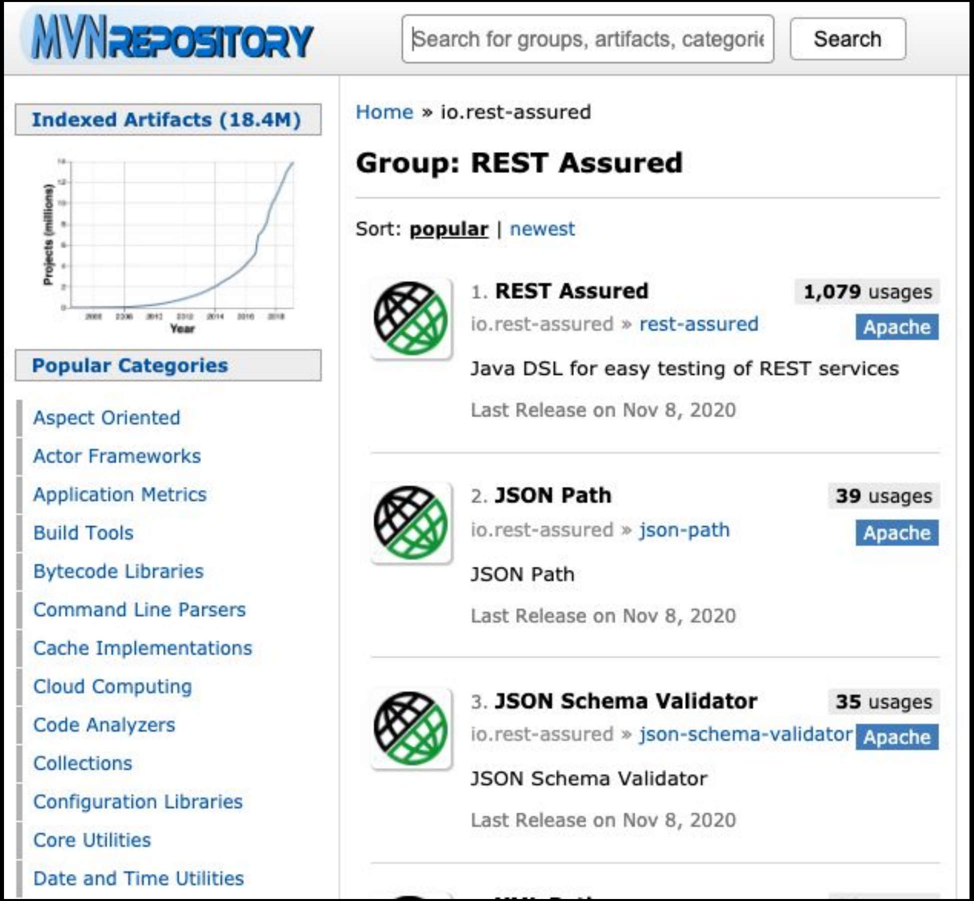
REST Assured is a Java DSL for simplifying testing of REST based services built on top of HTTP Builder.

It supports POST, GET, PUT, DELETE, OPTIONS, PATCH and HEAD requests and can be used to validate and verify the response of these requests.

https://rest-assured.io/

https://github.com/rest-assured/rest-assured/wiki/Usage

Rest Assured


MVNREPOSITORY

Search for groups, artifacts, categories | Search

Indexed Artifacts (18.4M)

Popular Categories

- Aspect Oriented
- Actor Frameworks
- Application Metrics
- Build Tools
- Bytecode Libraries
- Command Line Parsers
- Cache Implementations
- Cloud Computing
- Code Analyzers
- Collections
- Configuration Libraries
- Core Utilities
- Date and Time Utilities

Home » io.rest-assured

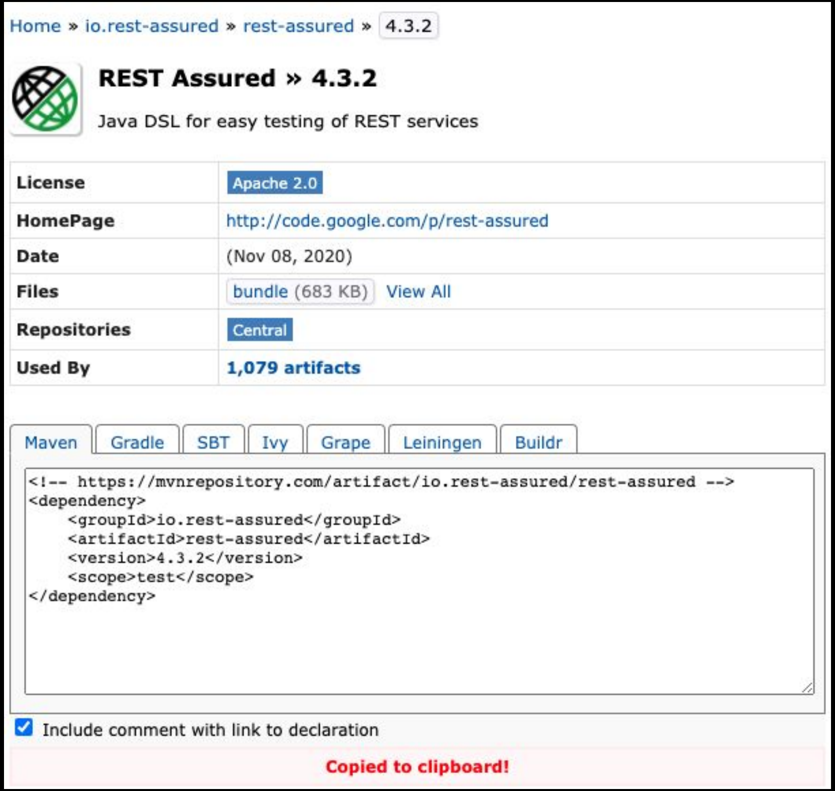
Group: REST Assured

Sort: **popular** | newest

1. **REST Assured** — **1,079** usages
   io.rest-assured » rest-assured — Apache
   Java DSL for easy testing of REST services
   Last Release on Nov 8, 2020

2. **JSON Path** — **39** usages
   io.rest-assured » json-path — Apache
   JSON Path
   Last Release on Nov 8, 2020

3. **JSON Schema Validator** — **35** usages
   io.rest-assured » json-schema-validator — Apache
   JSON Schema Validator
   Last Release on Nov 8, 2020


Home » io.rest-assured » rest-assured » 4.3.2

REST Assured » 4.3.2

Java DSL for easy testing of REST services

License	Apache 2.0
HomePage	http://code.google.com/p/rest-assured
Date	(Nov 08, 2020)
Files	bundle (683 KB) View All
Repositories	Central
Used By	**1,079 artifacts**

Maven | Gradle | SBT | Ivy | Grape | Leiningen | Buildr

```
<!-- https://mvnrepository.com/artifact/io.rest-assured/rest-assured -->
<dependency>
    <groupId>io.rest-assured</groupId>
    <artifactId>rest-assured</artifactId>
    <version>4.3.2</version>
    <scope>test</scope>
</dependency>
```

[✓] Include comment with link to declaration

Copied to clipboard!

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

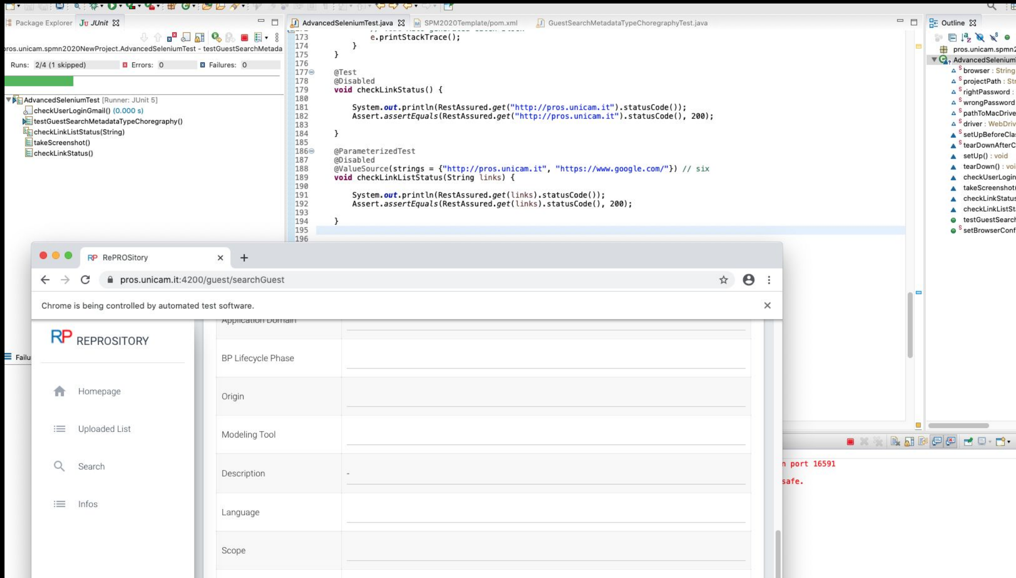
Explore testGuestSearchMetadataTypeChoregraphy method

Katalon https://www.katalon.com/

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?



```
            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);

    }
```

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

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

pros.unicam.it:4200/guest/searchGuest

Chrome is being controlled by automated test software.

Headless Browser…

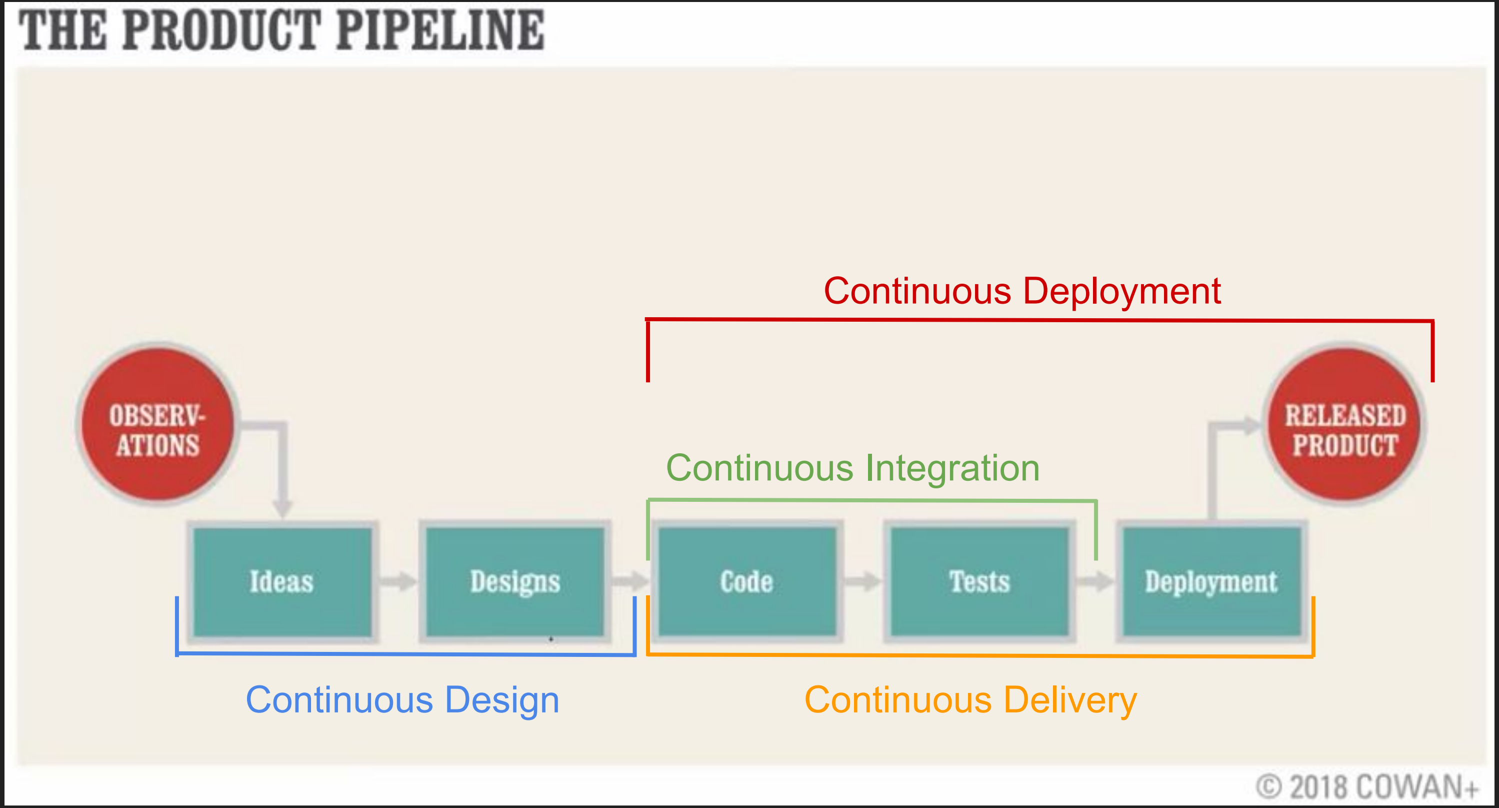
Headless Browser…

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

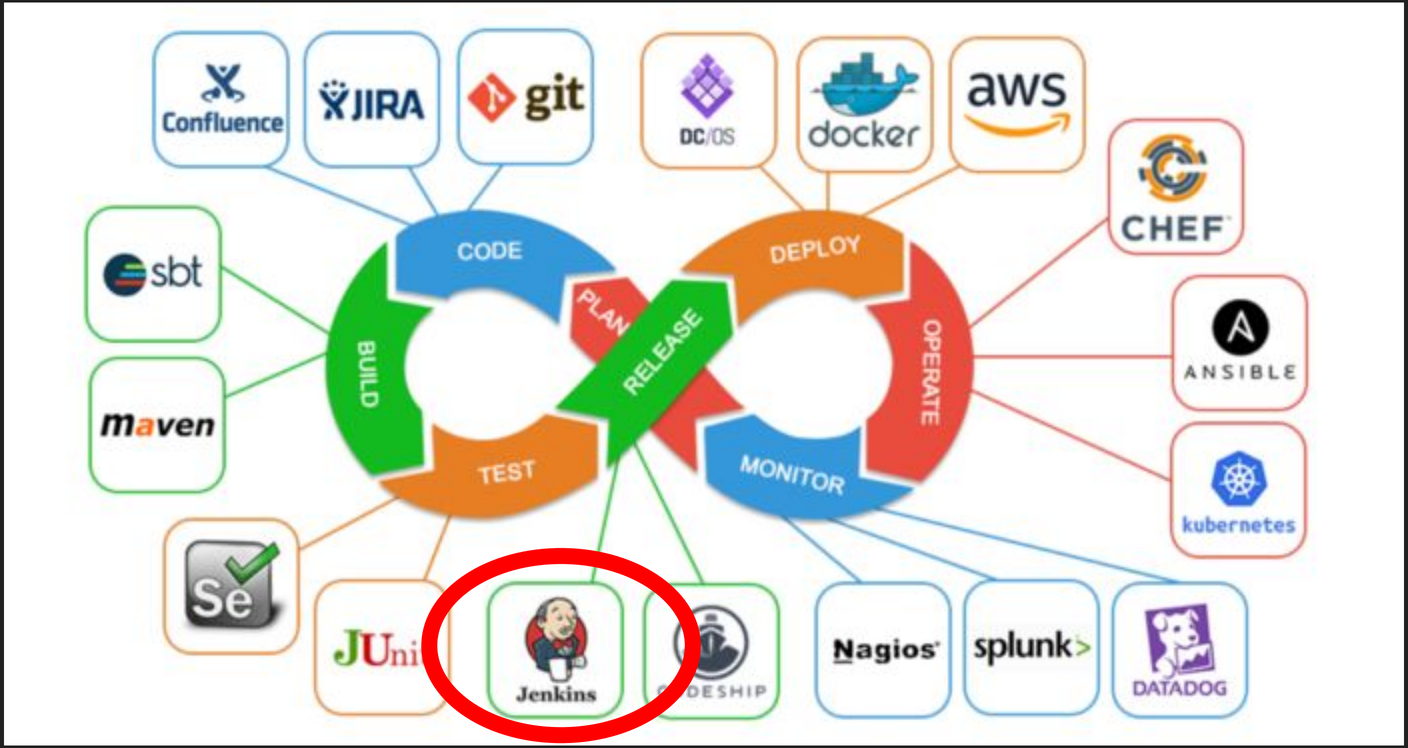
The Product Pipeline



Continuous Integration

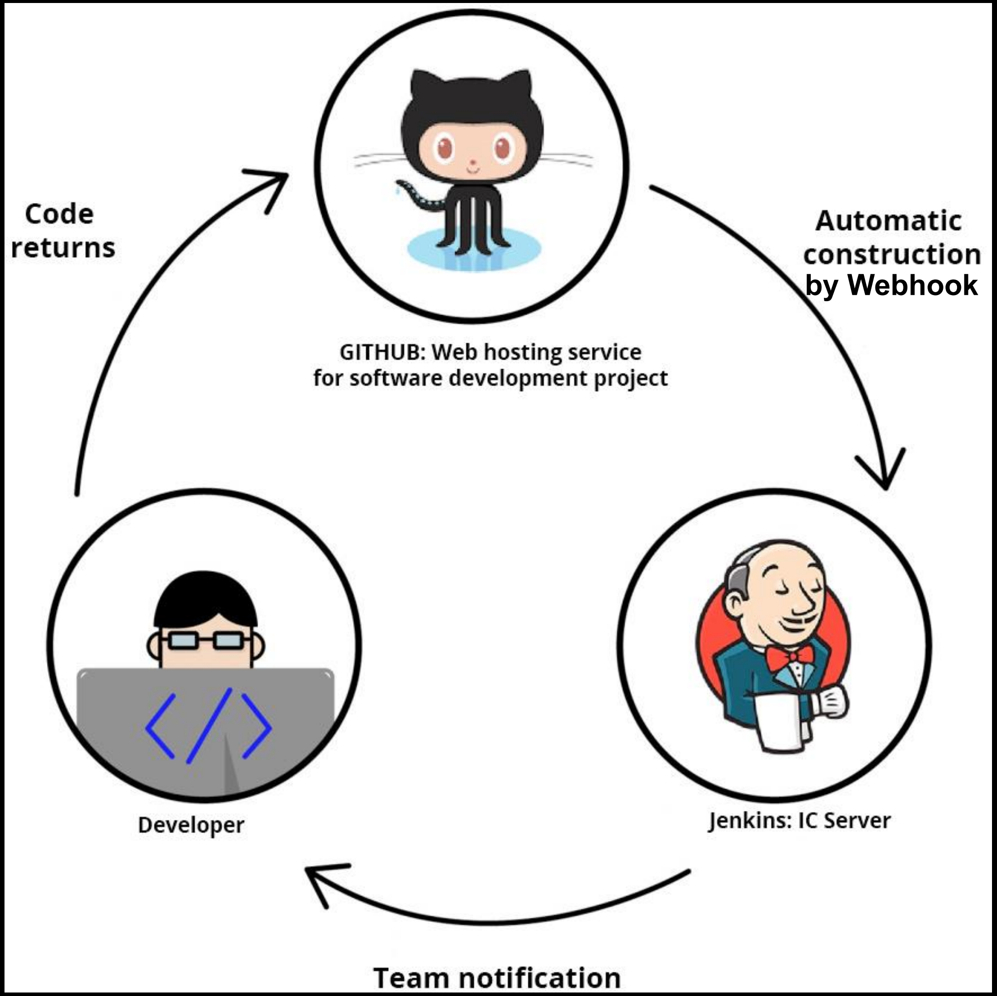
In its simplest form, it involves a tool that monitors your version control system for changes. Whenever a change is detected, this tool automatically compiles and tests your application. If something goes wrong, the tool immediately notifies the developers so that they can fix the issue immediately.

DevOps


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

Continuous Integration with Jenkins

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


Code returns
Automatic construction by Webhook
GITHUB: Web hosting service for software development project
Developer
Jenkins: IC Server
Team notification

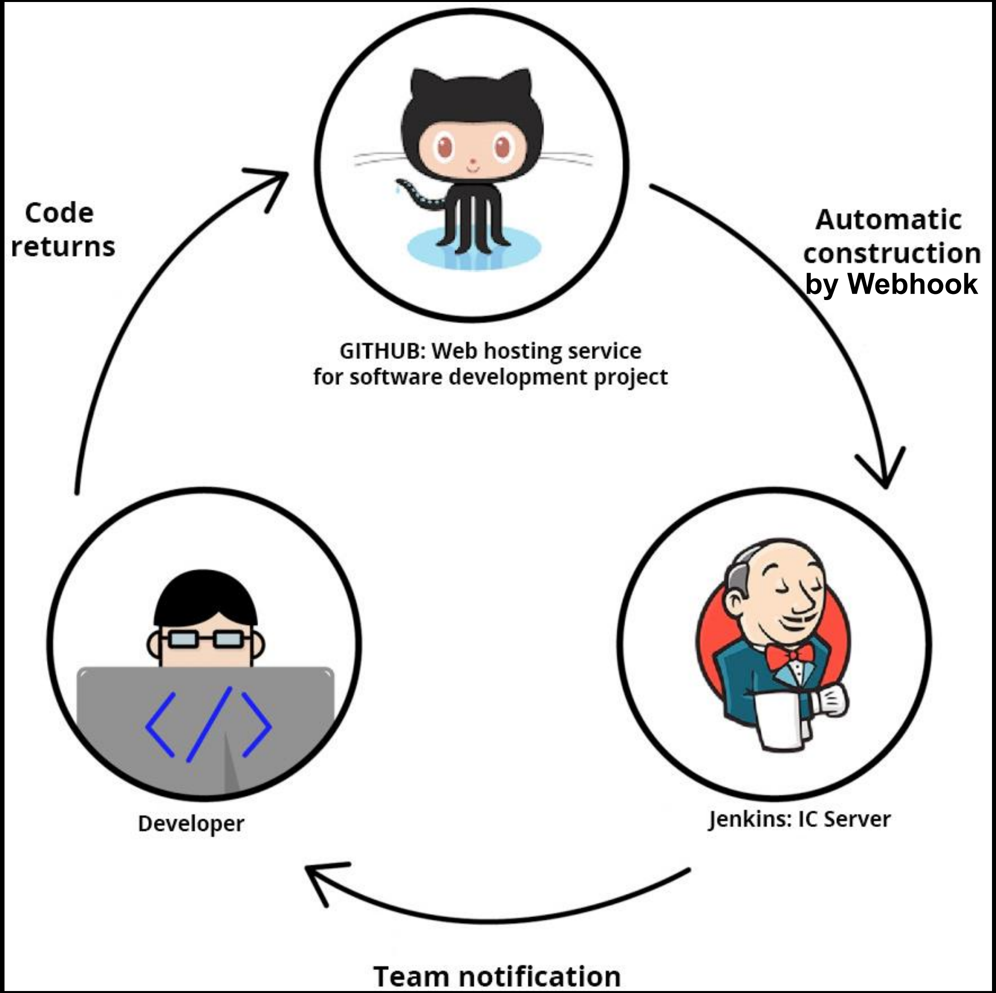
Jenkins

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

https://jenkins.io/

Continuous Integration with Jenkins

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


Code returns
Automatic construction by Webhook
GITHUB: Web hosting service for software development project
Developer
Jenkins: IC Server
Team notification

Jenkins Plugins

https://plugins.jenkins.io/

Environments

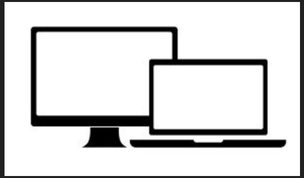
NOTE: Referred also as Development, Testing, Acceptance and Production (DTAP)

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

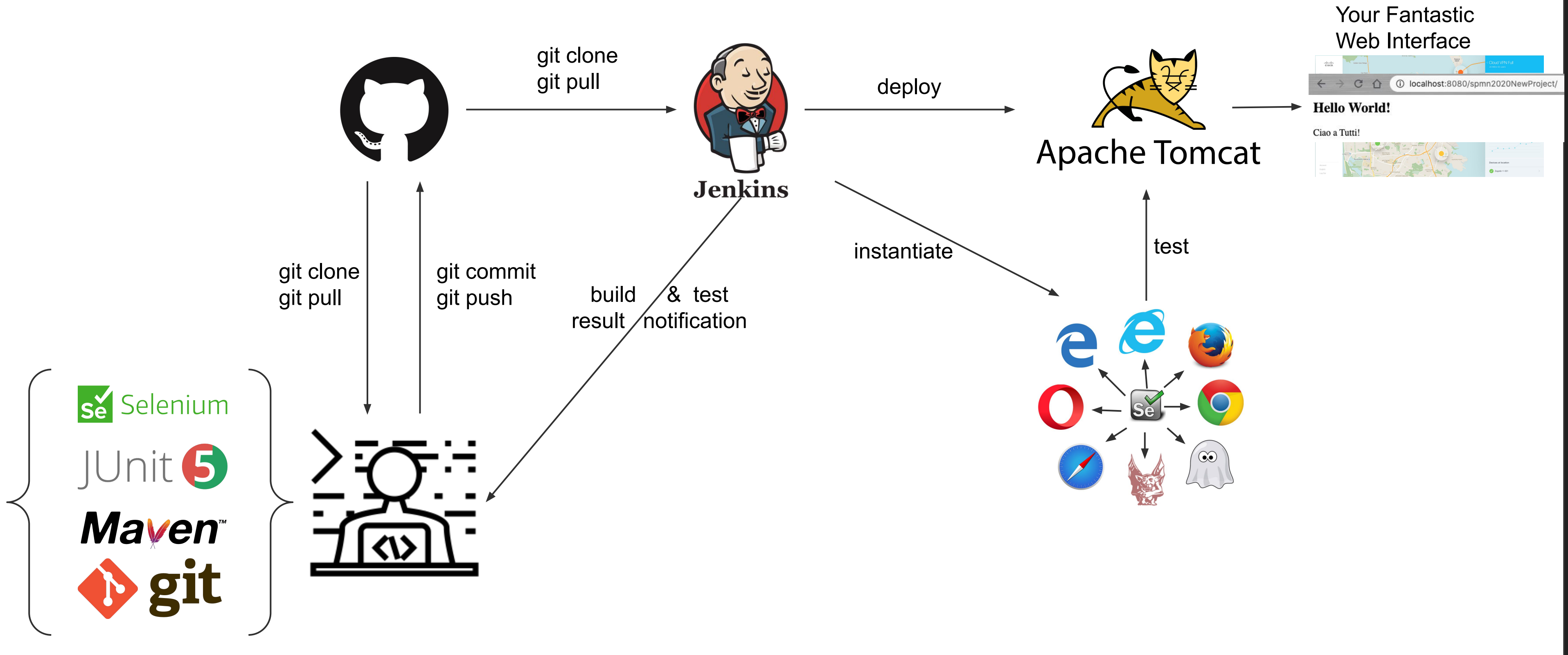
The staging site is used to assemble, test and review new versions of a web app before it goes into production.

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.

Our DevOps Toolchain



Not only Jenkins

TRAVIS CI

https://travis-ci.org/

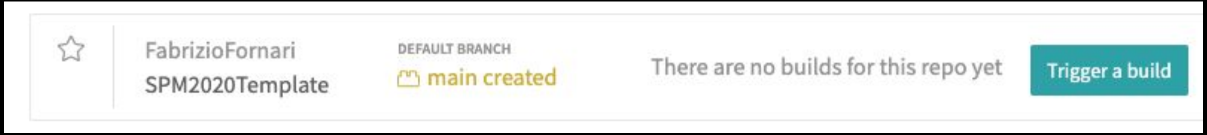
Travis CI

Travis CI is a hosted, distributed continuous integration service used to build and test software projects hosted at GitHub.

Travis CI is configured by adding a file named .travis.yml, which is a YAML format text file, to the root directory of the repository.

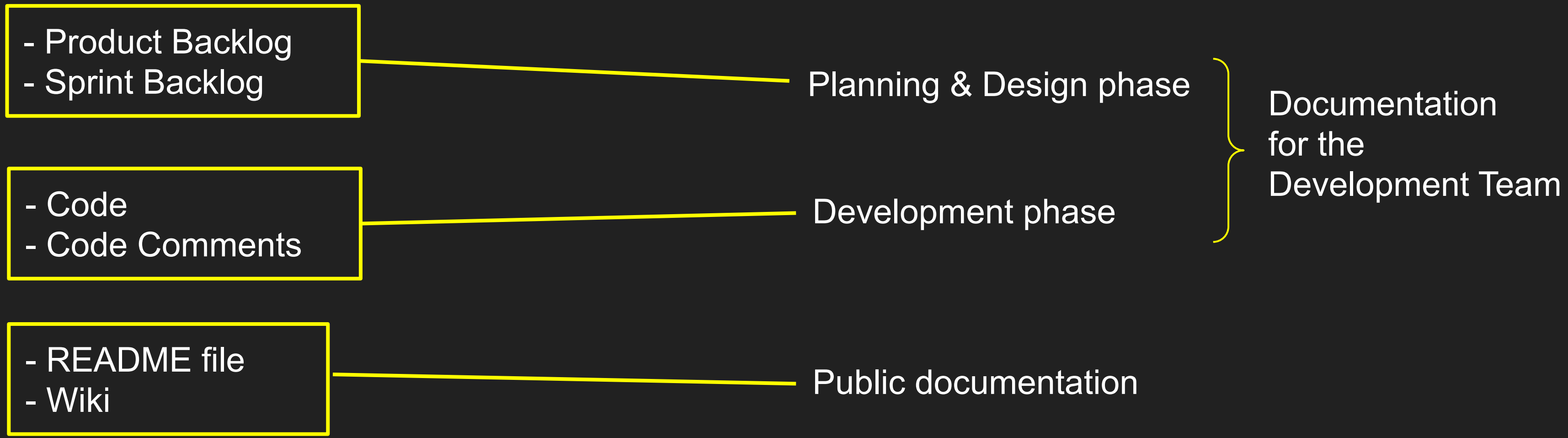
```
# this is a java project using maven
language: java
# install
install: mvn install
```

https://travis-ci.org/


FabrizioFornari
SPM2020Template
DEFAULT BRANCH
main created
There are no builds for this repo yet

Travis CI Documentation: https://docs.travis-ci.com/

Documentation in an Agile/Scrum project



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

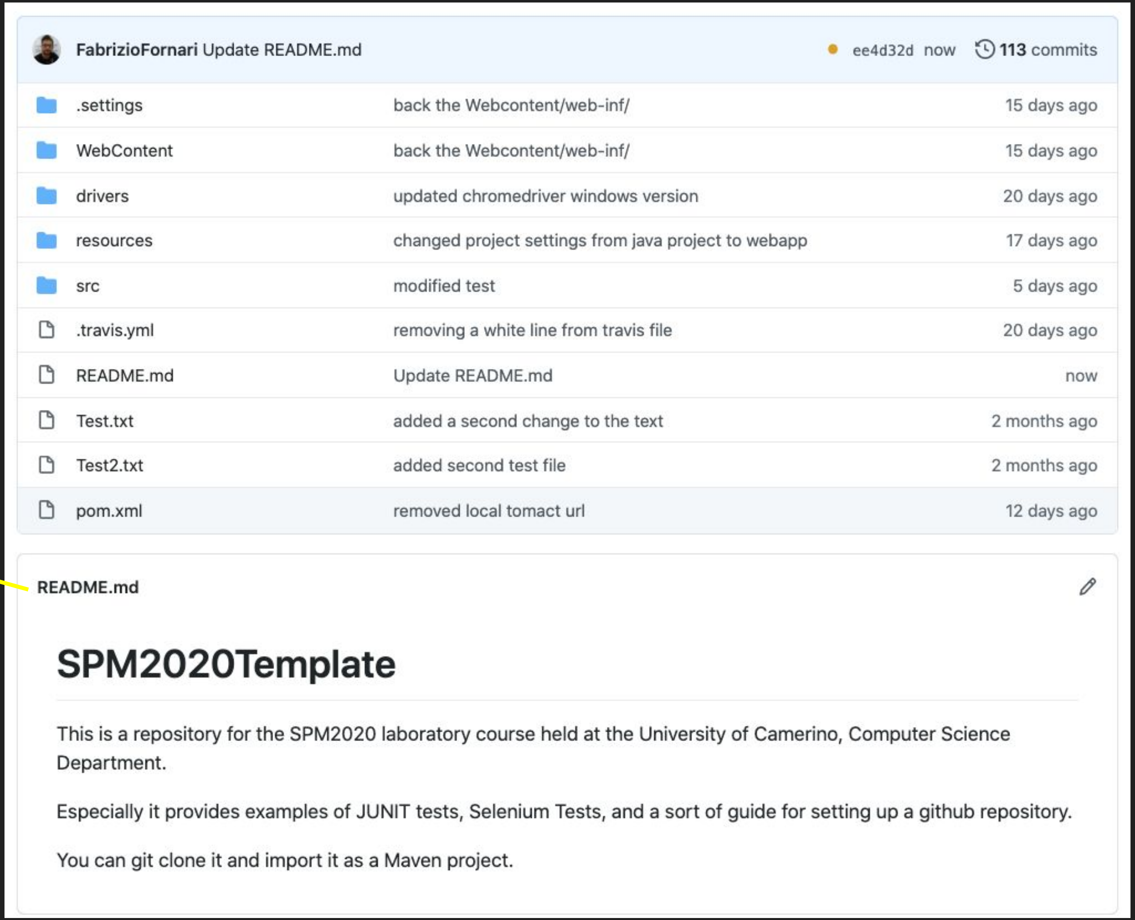
A README is often the first item a visitor will see when visiting your repository. README files typically include information on:

- 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.

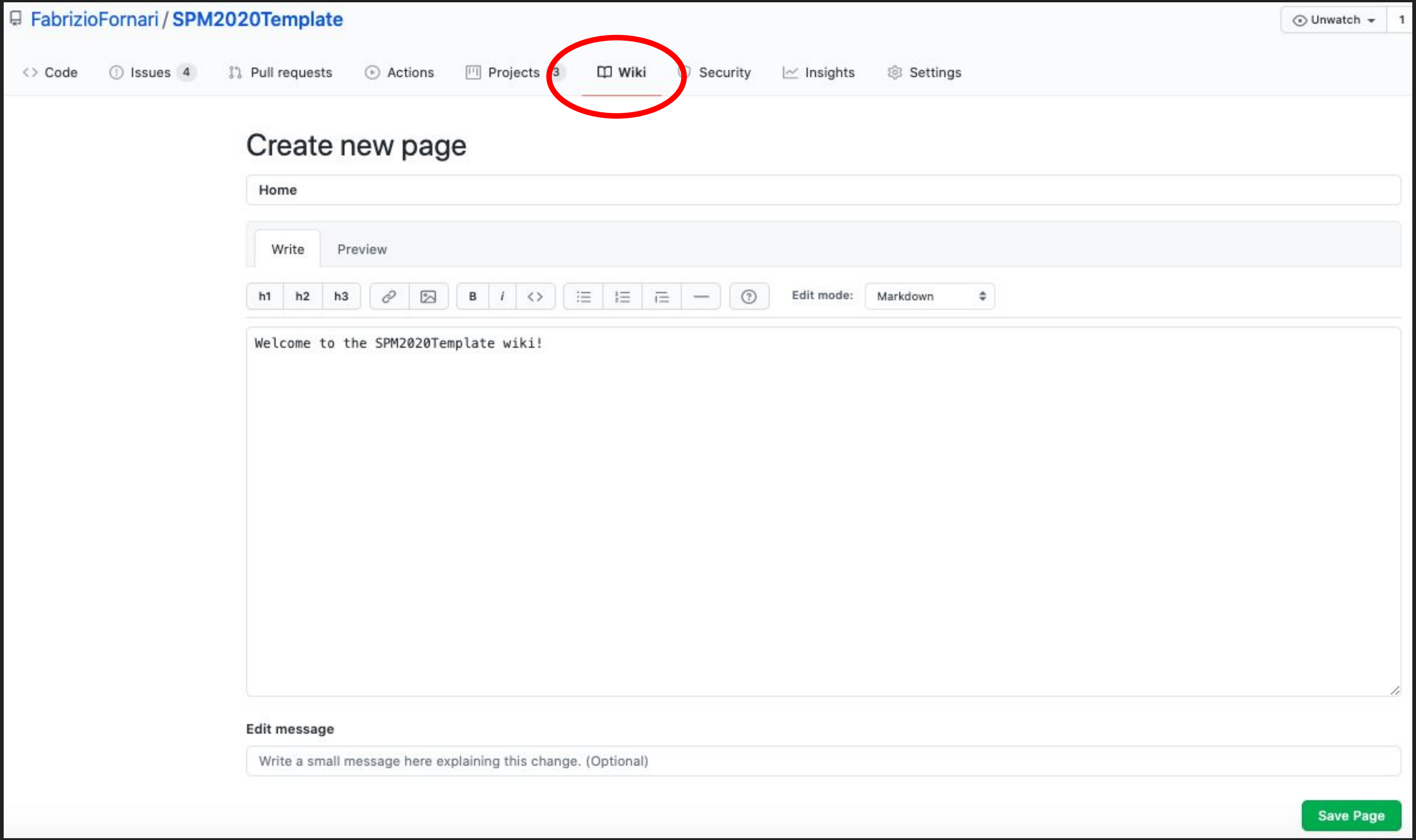
If you create a wiki in a public repository, the wiki is available to the public. If you create a wiki in an internal or private repository, people with access to the repository can also access the wiki.

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

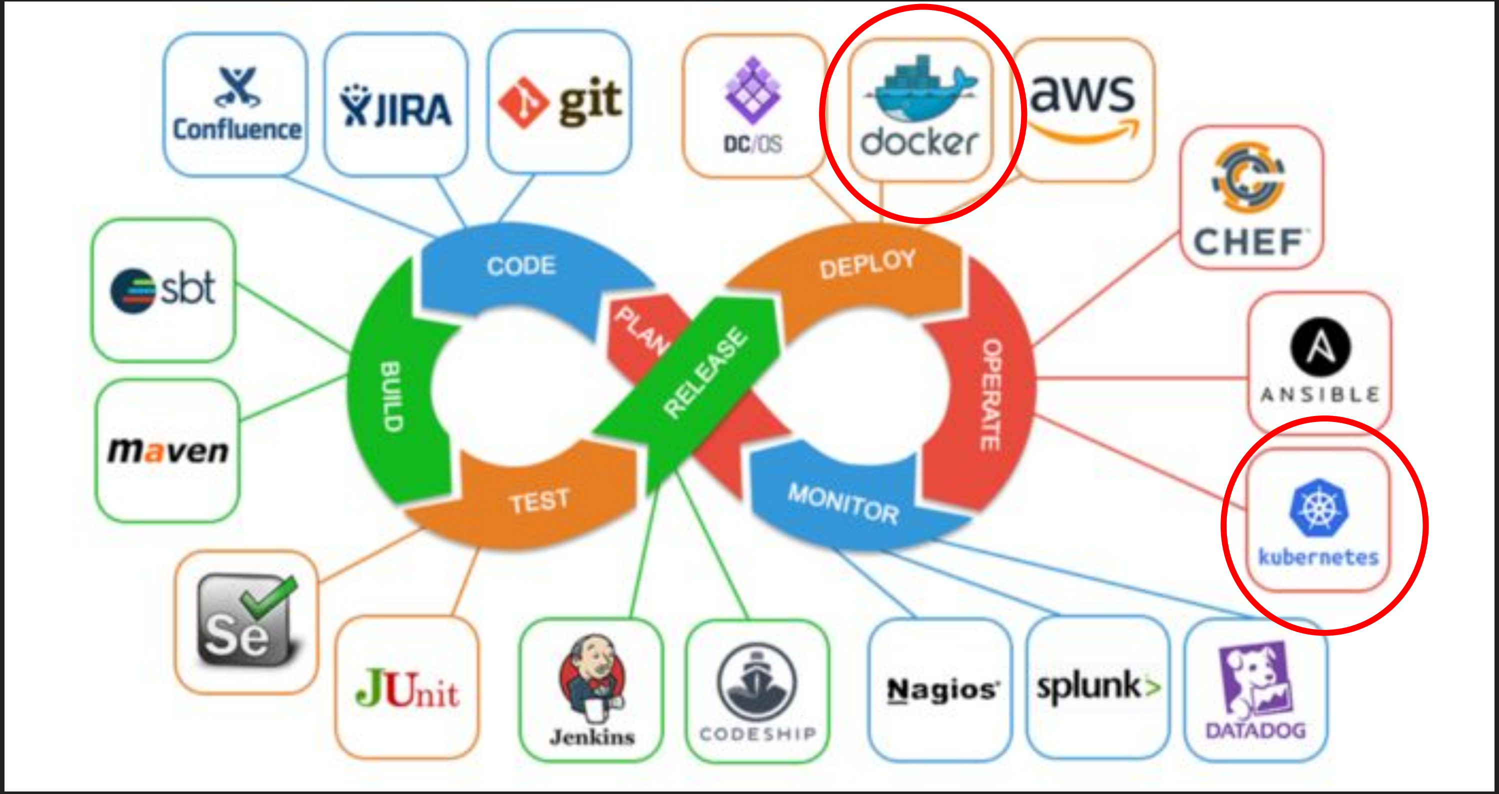
```
Welcome to the SPM2020Template wiki!
```

Edit message

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

Looking ahead...

DevOps Technologies



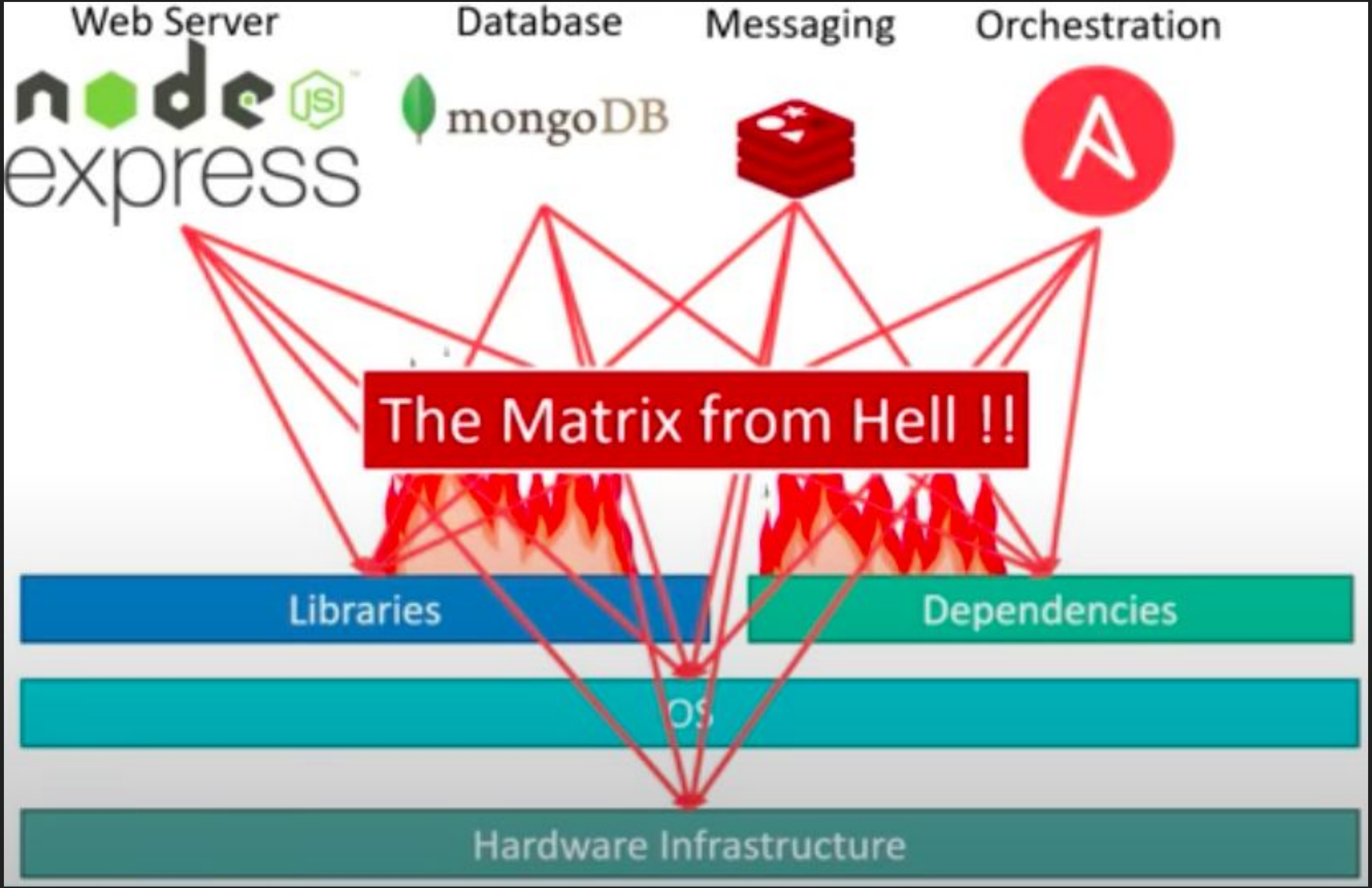
https://www.docker.com/

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.

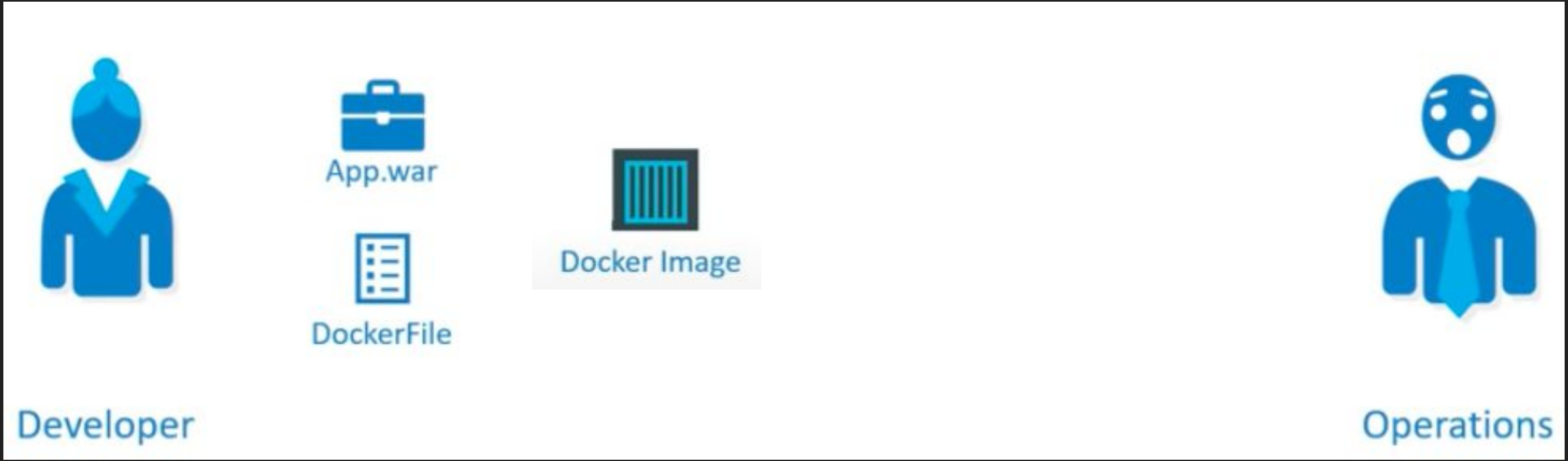
Compatibility/
Dependency

Long setup
time

Different
Dev/Test/Prod
environments



From Application to Container



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

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

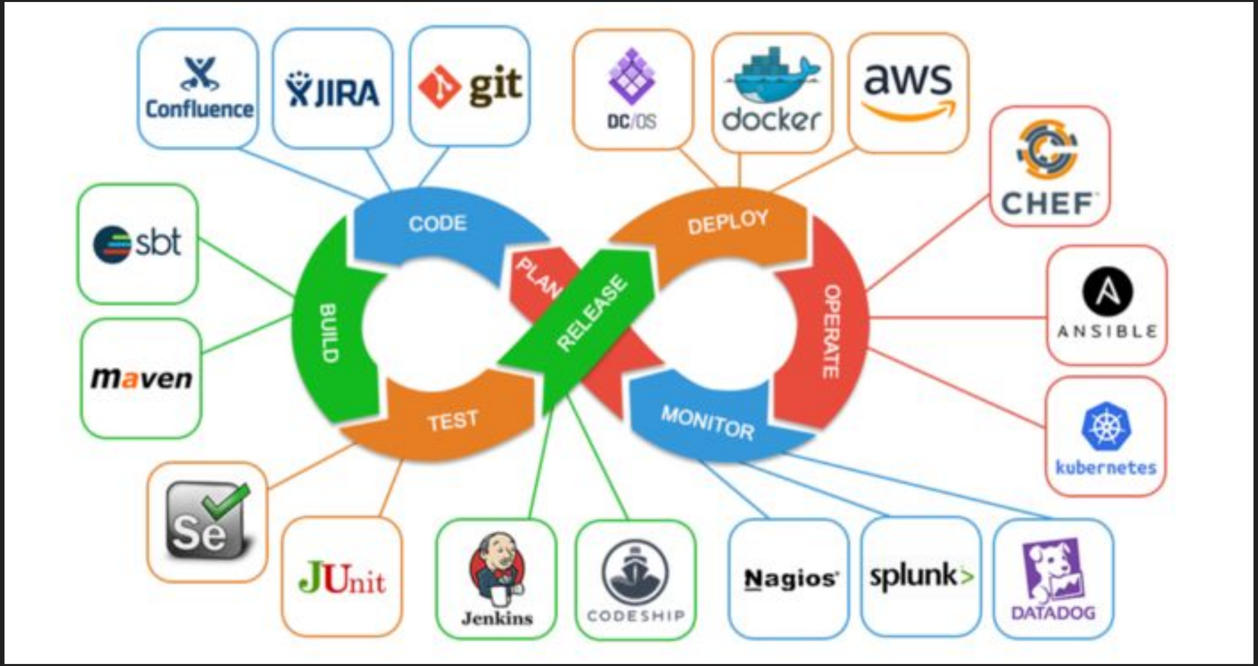
Kubernetes: from Google, difficult to setup but supports many advanced features, all public cloud supports it

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.

What’s next?



PROS Lab
pros.unicam.it

My Topics

Business Process Management
Business Process Modeling and Verification
Business Process and IoT
Process Mining
Software Project Management

Tools:

- BProVe, Business Process Verifier
- BEBoP, understandaBility vErifier for Business Process models
- RePROSitory, Repository of open PROcess modelS

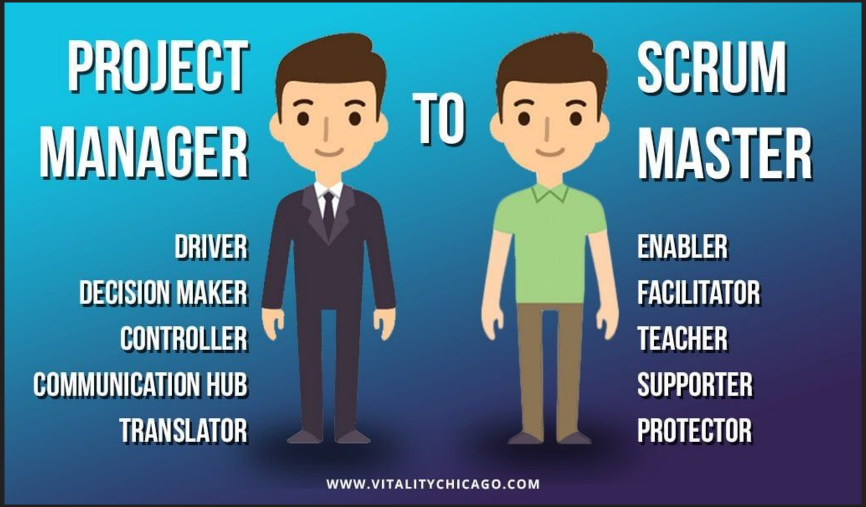
Group Projects or Thesis

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.



Questionario Online

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