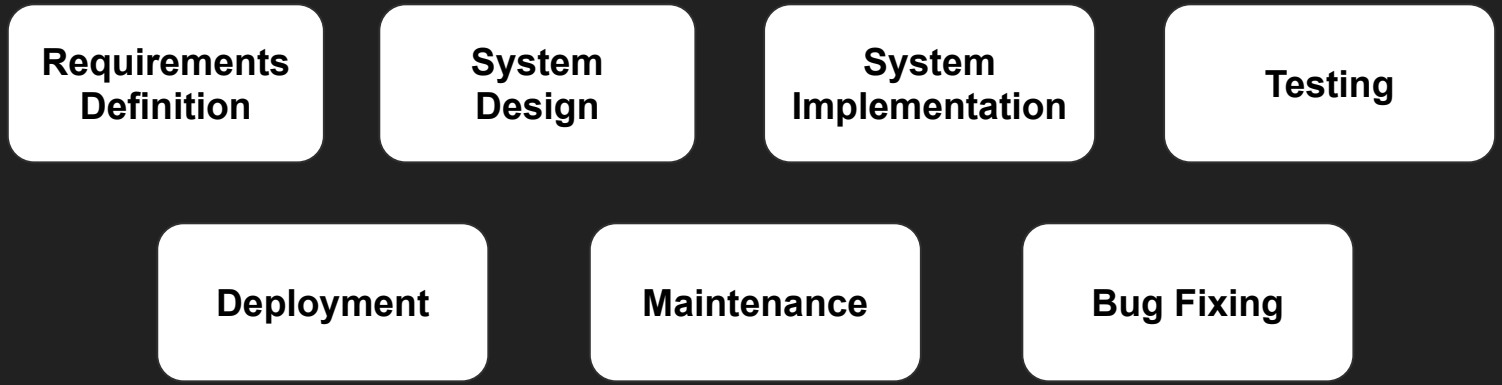
Software Project Management - Laboratory

Lecture n° 19
A.Y. 2021-2022

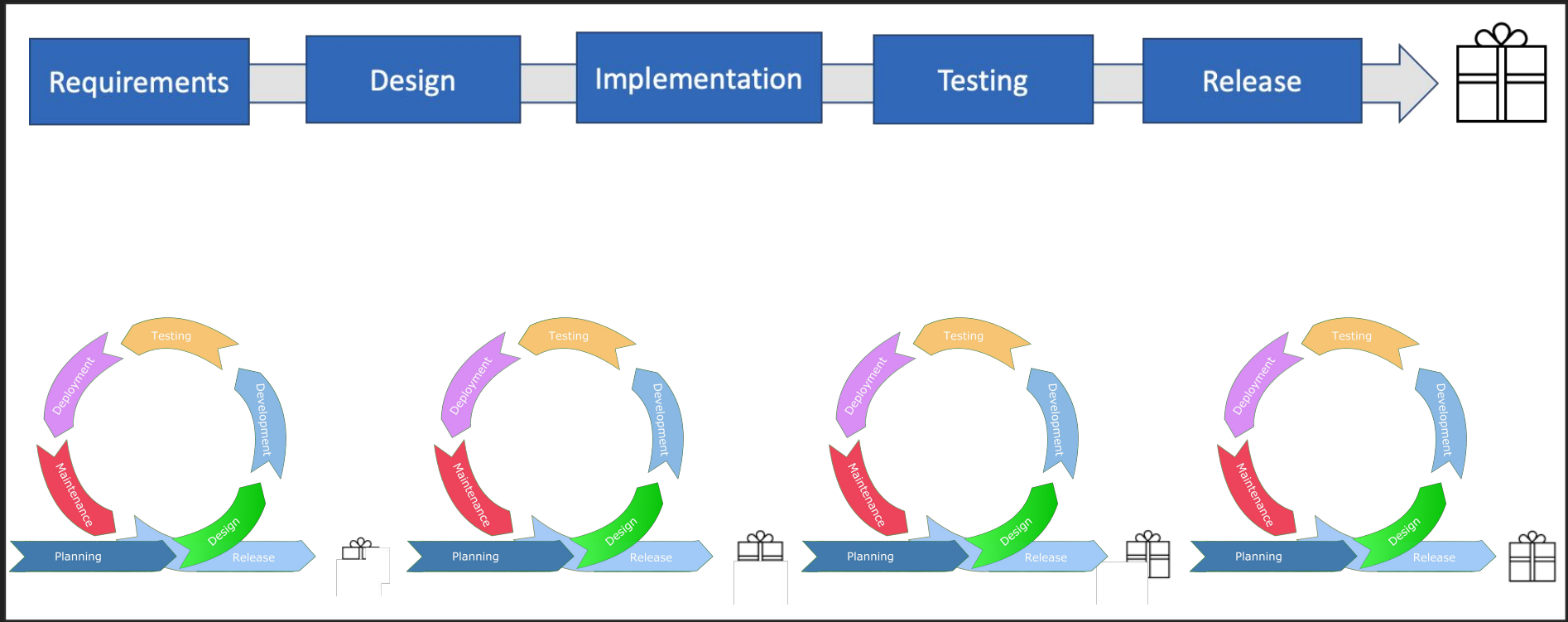
Prof. Fabrizio Fornari

Software Development Process

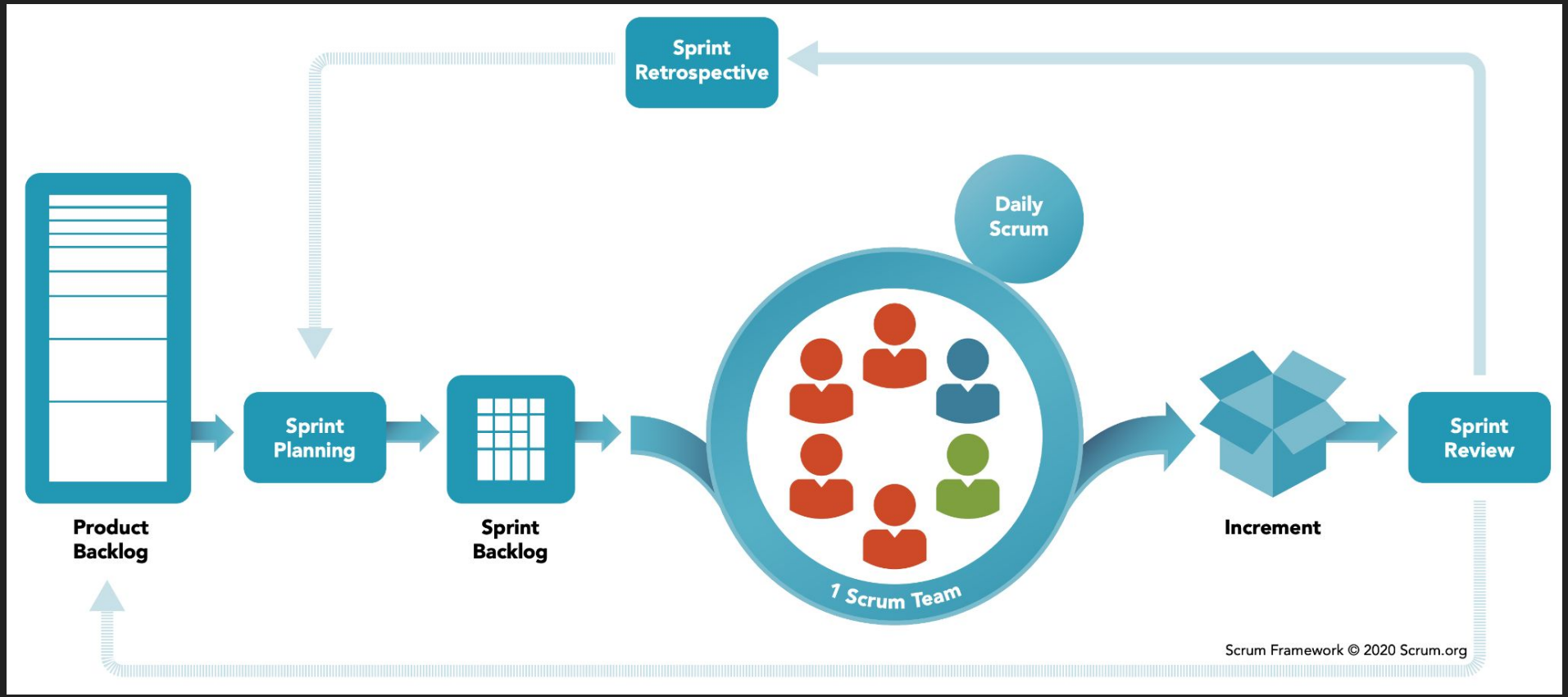
Software Development Process is the process of dividing software development work into distinct phases to improve design, product management, and project management. It is also known as a software development life cycle (SDLC)



Waterfall vs Agile



SCRUM - Framework



Environments

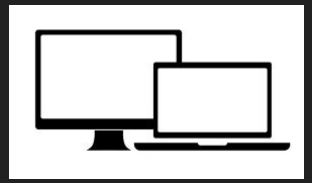
NOTE: Referred also as Development, Testing, Acceptance and Production (DTAP)

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

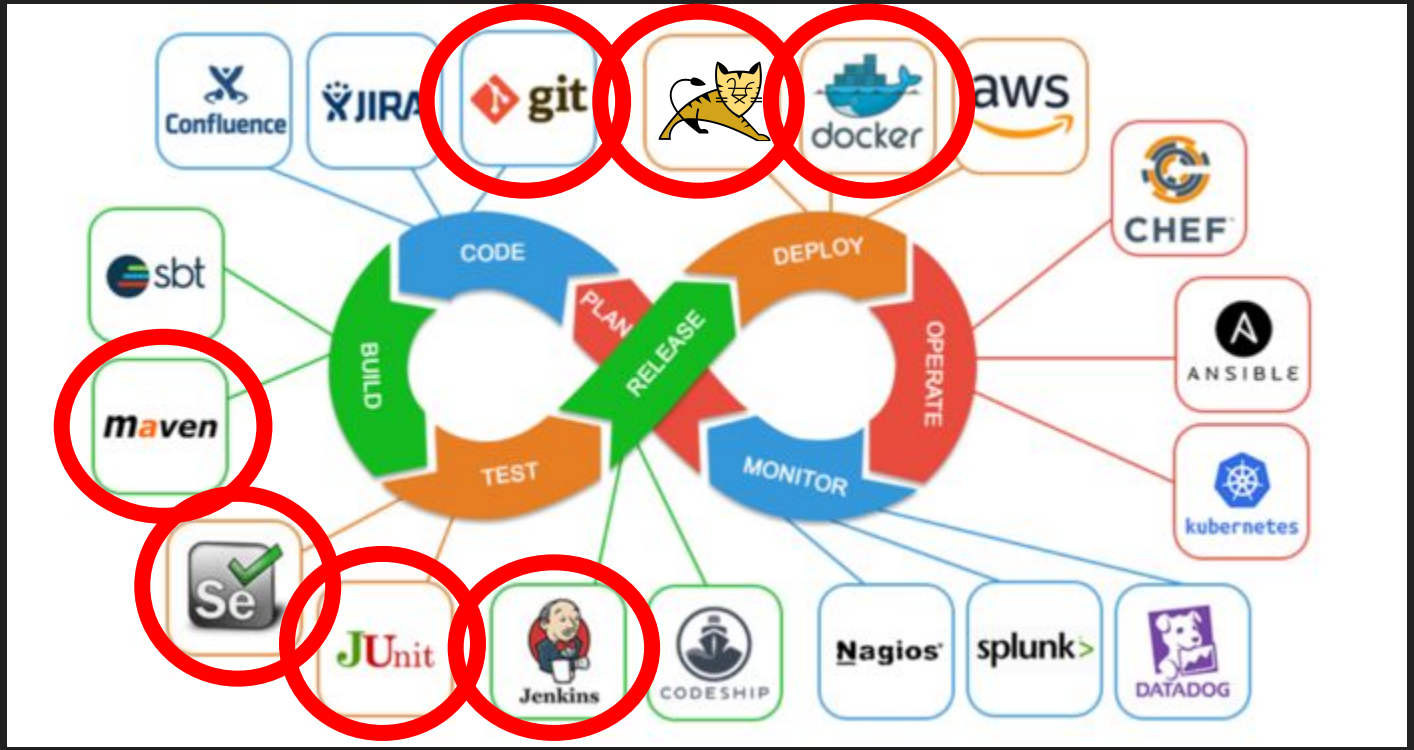
The staging site is used to assemble, test and review new versions of a web app before it goes into production.

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

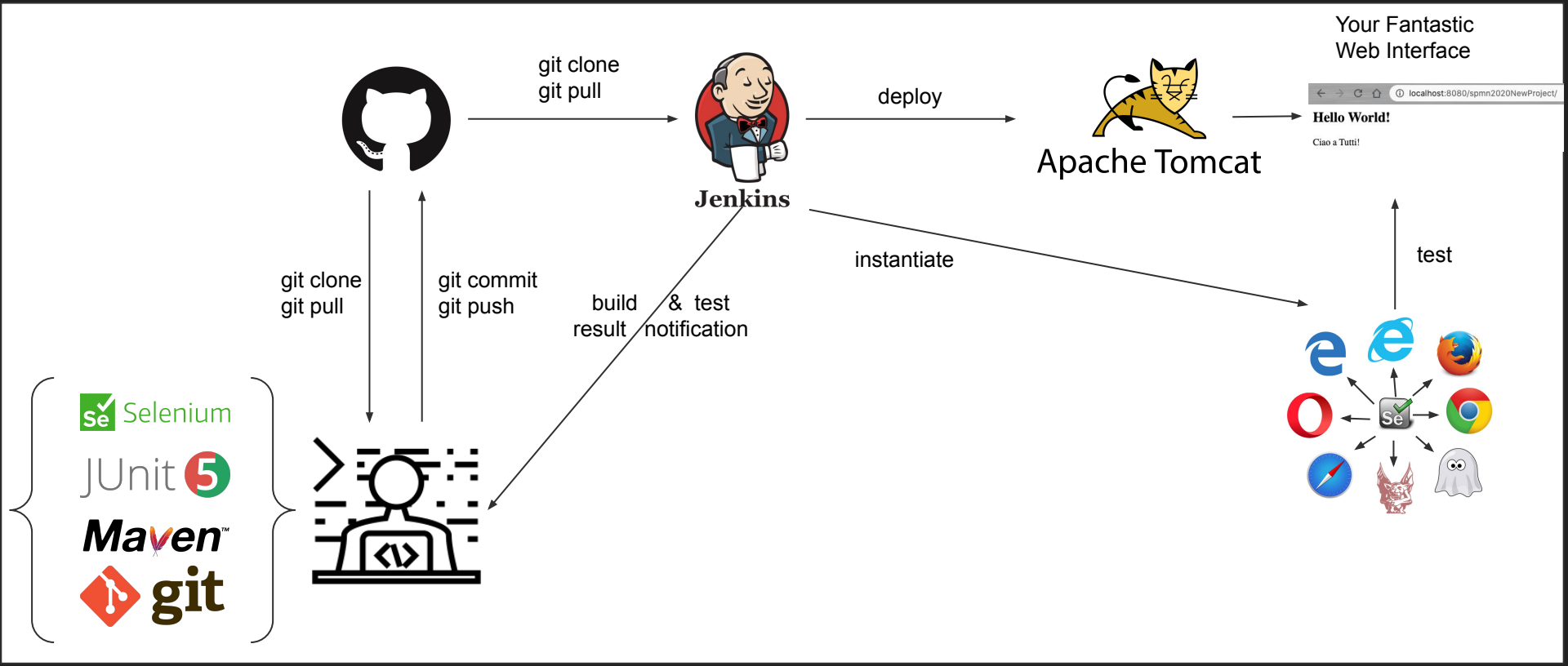
Production

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

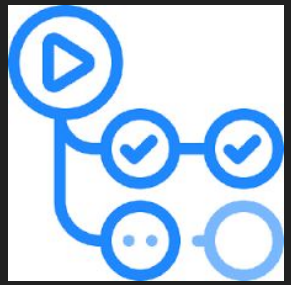
DevOps


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

Our DevOps Toolchain



GitHub Actions



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

Get executed on GitHub Server

Jobs are execute on virtual machines hosted by GitHub.

FabrizioFornari / spm2021Template Public

Unwatch 1 Star 0 Fork 18

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

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

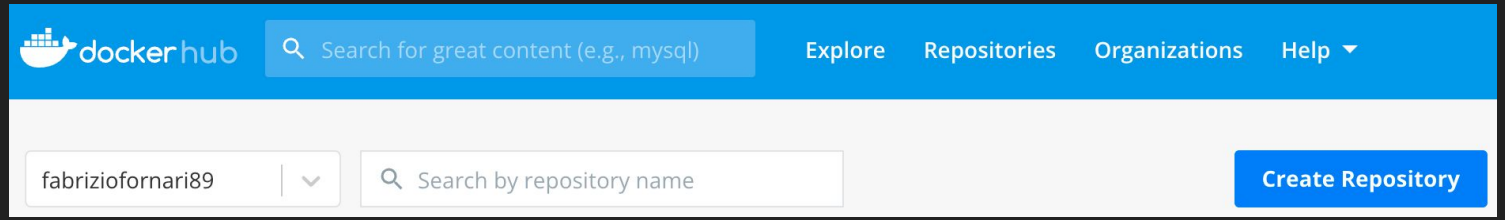
Deploy to Amazon ECS
By Amazon Web Services
Deploy a container to an Amazon ECS service powered by AWS Fargate or Amazon EC2.
Set up this workflow

actions/starter-workflows Deployment

Public Docker Images Repository

https://hub.docker.com/

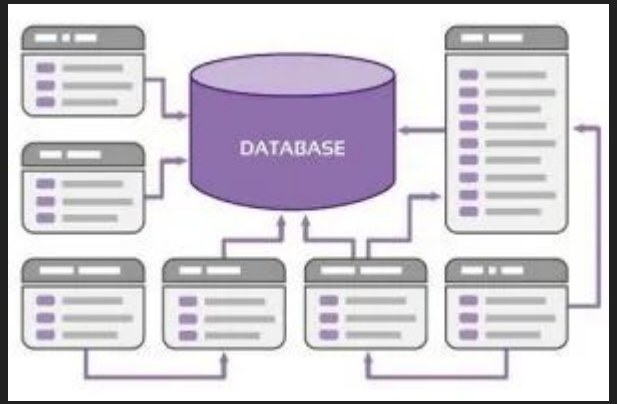
Create an account and a Private Repository



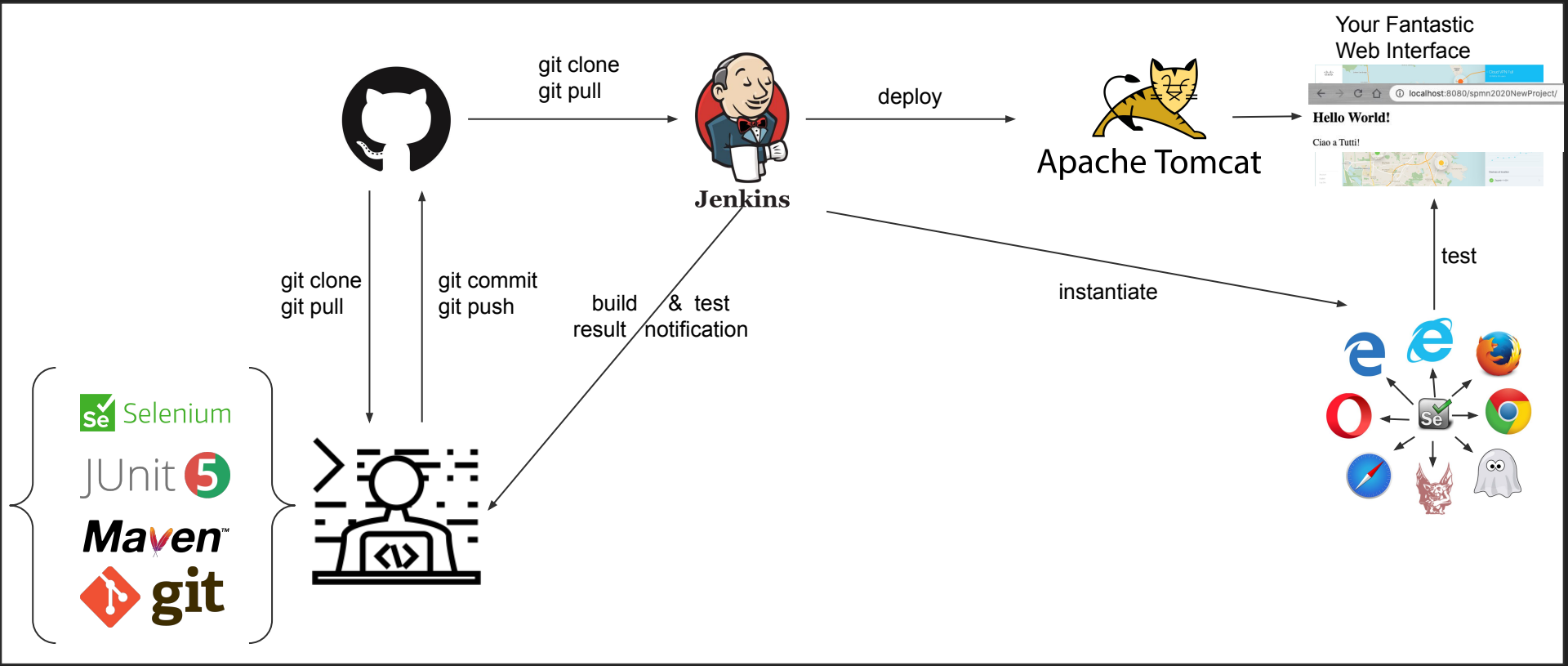
Database

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.



How Databases fit into CI/CD?



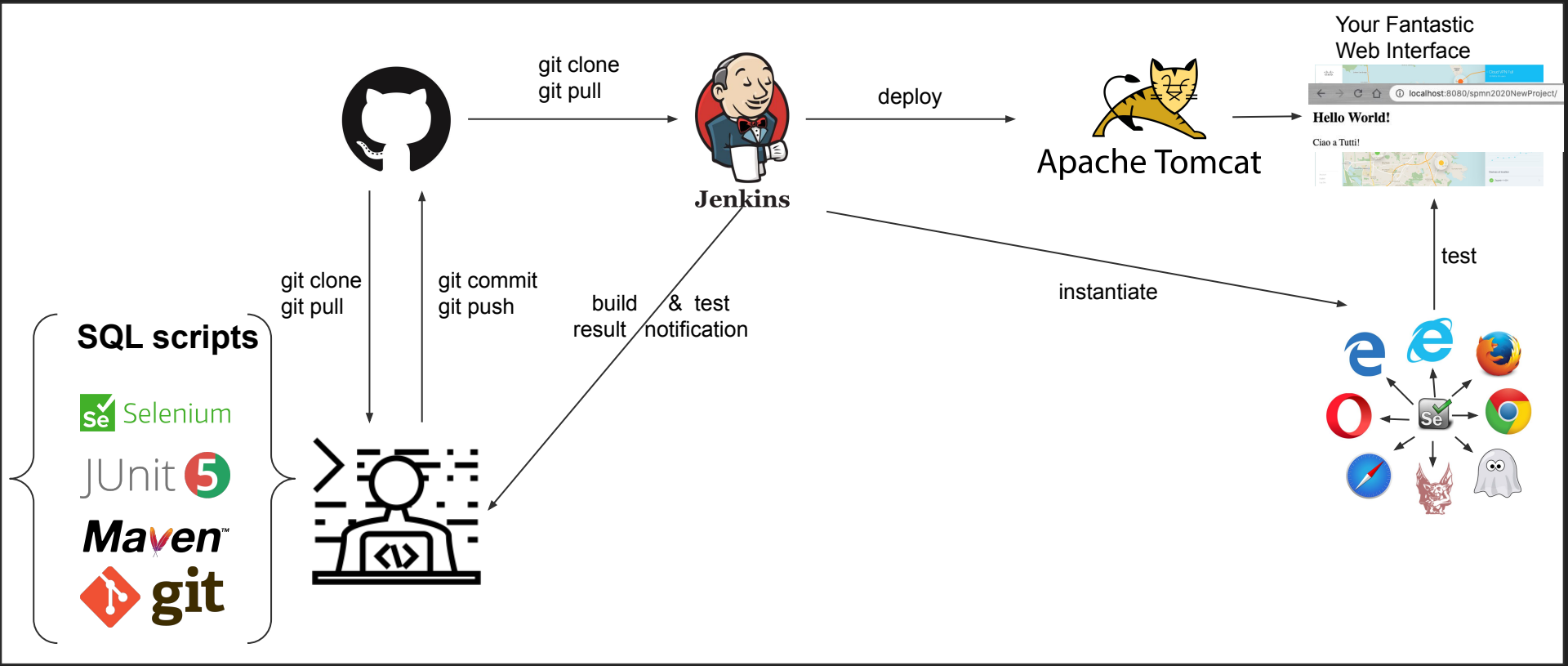
Special Guest

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.

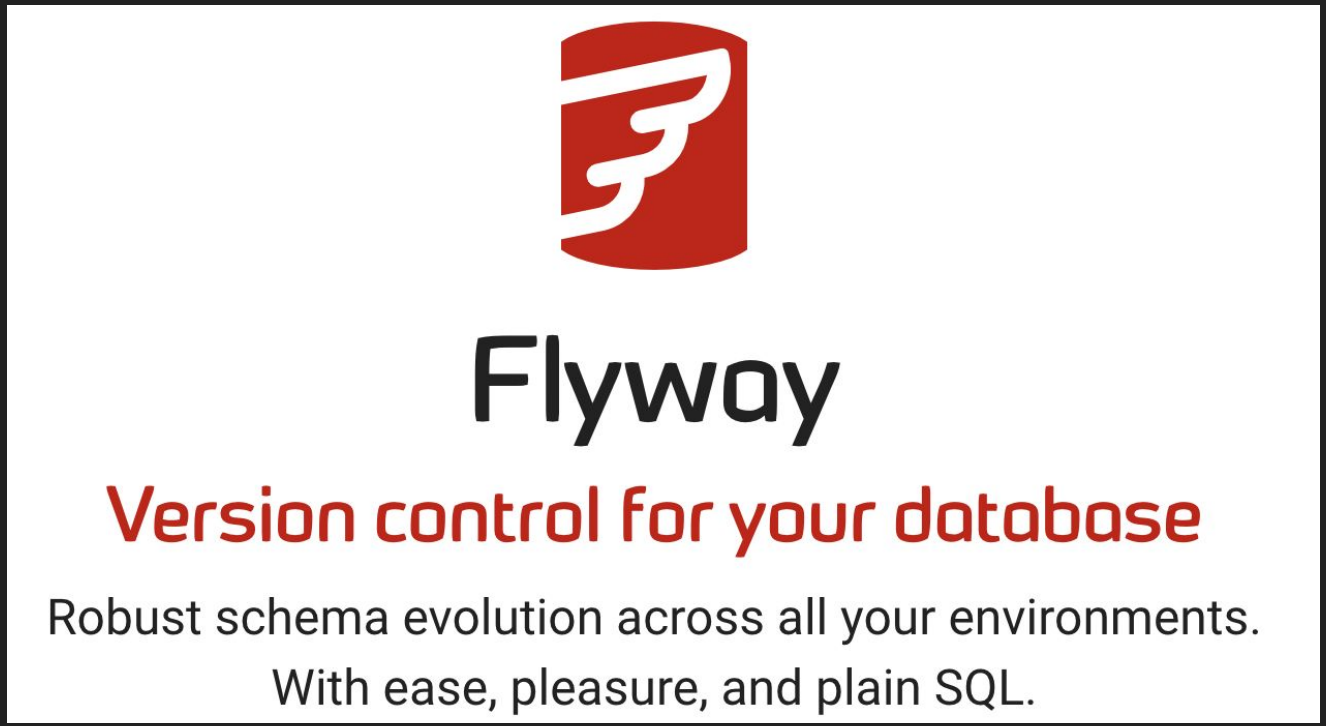
She’s currently writing her Master Thesis on the topic of Database Schema Evolution and Testing during Continuous Integration.

How Databases fit into CI/CD?



How Databases fit into CI/CD?

Especially regarding Relational Database we can speak about Database Migration



Flyway

Flyway is an open-source tool, licensed under Apache License 2.0, that helps implementing automated and version-based database migrations.

It allows to define the required update operations in an SQL script or as Java code.

We can then run the migration from a command line client or automatically as part of a build process or integrated into a Java application.

Exercise

1. Install PostgreSQL
2. Install pgAdmin
3. Clone https://github.com/FabrizioFornari/db_flyway_sample
4. Create a DB and a User flywaydemo (run createDatabaseAndUser.sh)
5. Execute mvn **clean compile flyway:migrate**
6. https://flywaydb.org/documentation/database

Configuration File

pom.xml

```
<properties>
    <properties>
        <flyway.version>8.2.2</flyway.version>
        <postgres.driver.version>9.1-901-1.jdbc4</postgres.driver.version>
        <database.url>jdbc:postgresql://localhost:5432/flywaydemo</database.url>
        <database.user>flywaydemo</database.user>
        <database.password>flywaydemo</database.password>
    </properties>
</properties>

<!-- https://mvnrepository.com/artifact/org.flywaydb/flyway-core -->
<dependency>
    <groupId>org.flywaydb</groupId>
    <artifactId>flyway-core</artifactId>
    <version>8.2.1</version>
</dependency>

<dependency>
   <groupId>postgresql</groupId>
    <artifactId>postgresql</artifactId>
     <version>9.1-901-1.jdbc4</version>
</dependency>
```

flyway.conf

```
flyway.driver=org.postgresql.Driver
flyway.url=jdbc:postgresql://localhost:5432/flywaydemo
flyway.user=flywaydemo
flyway.password=flywaydemo
flyway.locations=filesystem:src/main/resources/flyway/migrati
on_v1,filesystem:src/main/resources/flyway/migration_v2
flyway.sqlMigrationPrefix=V
flyway.sqlMigrationSeparator=__
flyway.sqlMigrationSuffix=.sql
flyway.validateOnMigrate=true
```

What’s next?

Date	Topic
17/12/2021	Sprint Meeting/ Project Status Check
23/12/2021	Review of the Entire Course
January	Sprint Review