Continuous Integration (CI)

David Afonso Dorta
Adexe Sabina Pérez

Index of the presentation

1. What is “Continuous Integration”
2. What is “Continuous Delivery”
3. Example setting a CI tool on a project
    a. Travis
    b. Codacy
    c. Git hooks
4. Bibliography

1

What is “Continuous Integration”

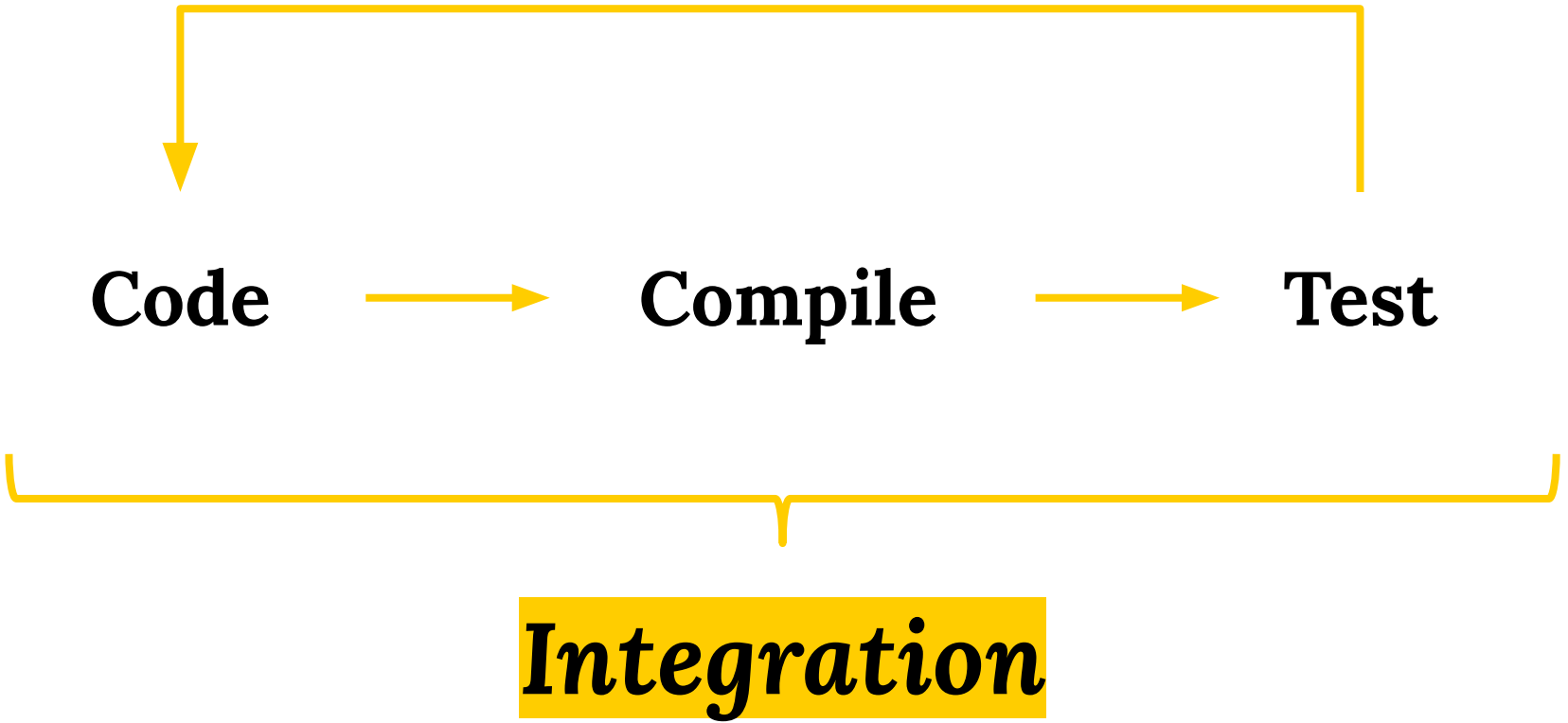
And why everybody talks about it

But first...

What does ***“Integration”*** *means?*

*The “**Integration**” process are all the required steps to **compile and test** a project.*

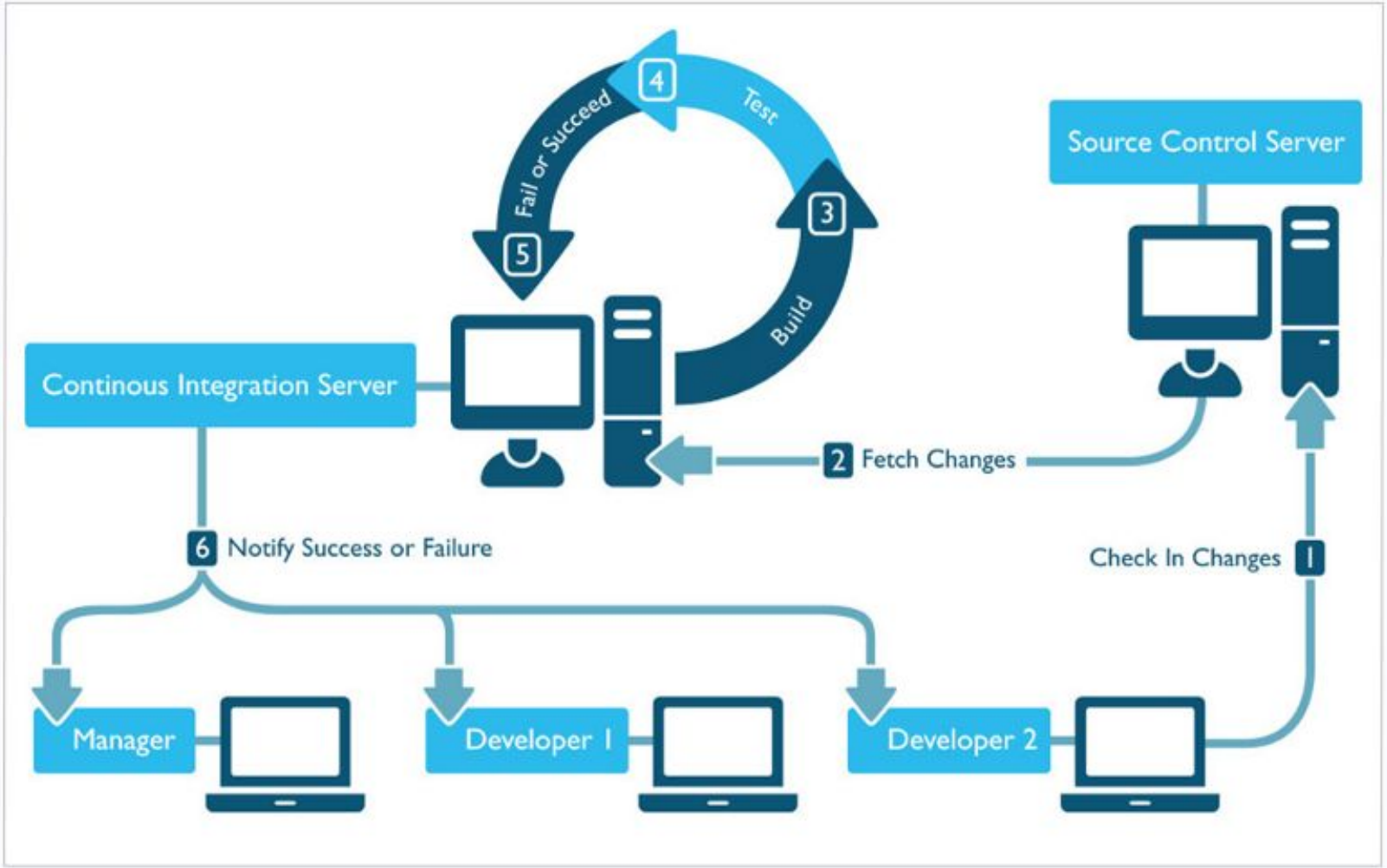
“


Code
Compile
Test
Integration

Continuous Integration

The process of **Integration**... **repeated over time.**

(As easy as it sounds)


Source Control Server
4
Test
3
Build
5
Fail or Succeed
Continous Integration Server
2 Fetch Changes
6 Notify Success or Failure
Check In Changes 1
Manager
Developer 1
Developer 2

Method created by

Martin Fowler

British Software Engineering specialized in **OOP design, UML, patterns, agile development...**

The idea

"The **frequency** of integration is **directly proportional** to the **easiness** of finding errors."

The idea

```
uppercase_sample = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
lowercase_sample = 'abcdefghijklmnopqrstuvwxyz'
digit_sample = '0123456789'
if keras.backend.image_dim_ordering() != 'tf' an
    keras.backend.set_image_dim_ordering('tf')
    print("INFO: '~/.keras/keras.json' sets 'ima
          "'th', temporarily setting to 'tf'")

# Create TF session and set as Keras backend ses
sess = tf.Session()
keras.backend.set_session(sess)

# Get MNIST test data
X_train, Y_train, X_test, Y_test = data_mnist(tr
                                              tr
                                              te
                                              te

assert Y_train.shape[1] == 10
```

Compile & Test (Integrate)

300 lines of code

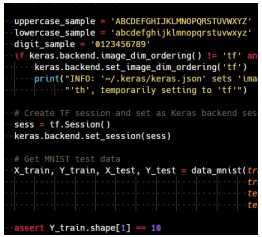
The idea

BAD PRACTICE

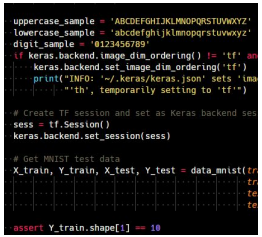
300 lines of code

Compile &
Test
(Integrate)

The idea



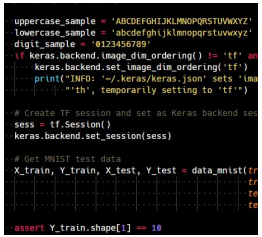
Integrate



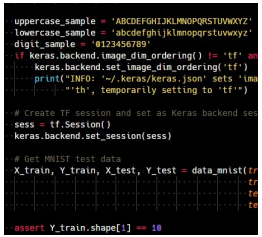
Integrate

50 lines of code

40 lines of code



Integrate

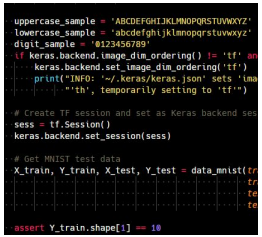


60 lines of code

30 lines of code

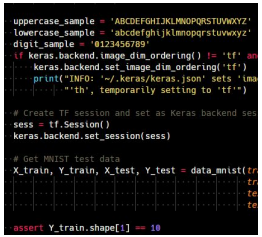
GOOD PRACTICE

The Rea



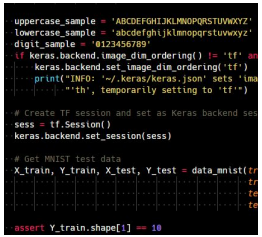
50 lines of code

Integrate



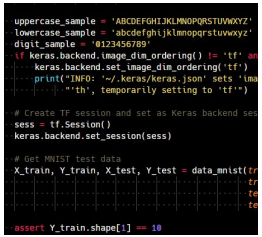
40 lines of code

Integrate



60 lines of code

Integrate

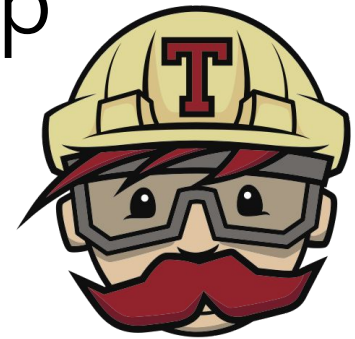


30 lines of code

CI can be tedious

Having to **constantly** perform the the integration of the build **manually** can **distract** a programmer from the actual work: ***write code***.

Codeship



circleci

CI/CD

Buddy

CI Automation Tools

Continuous Integration Keys

- **Automate** the build
- Make your build **self-testing**
- Every commit should build on an **integration machine**
- Test in a **clone** of the production environment
- Everyone can **see** what's happening

Team responsibilities

- Check in **frequently**
- Don’t check in **broken code**
- Don’t check in **untested code**
- Don’t check in **when the build is broken**
- **Don’t go home** after checking in **until the system builds**

2

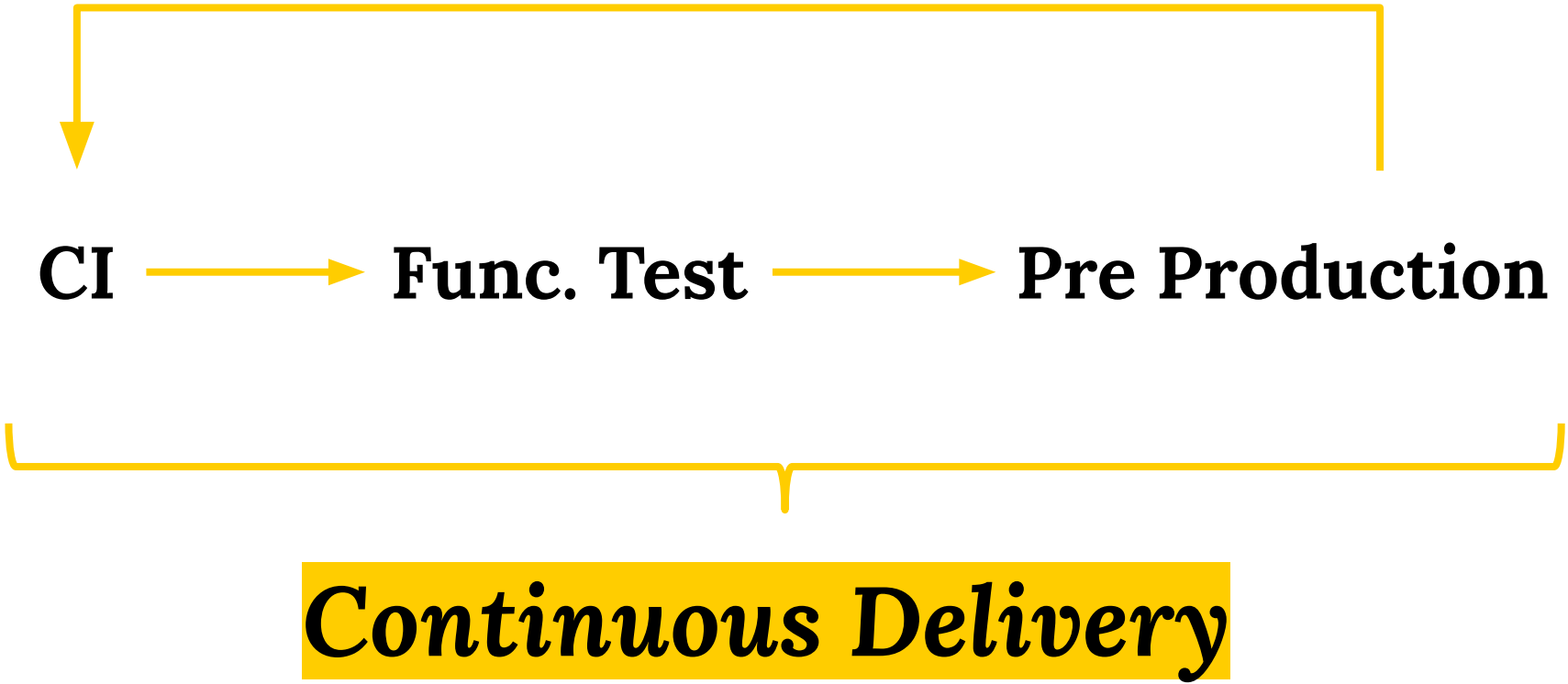
What is “Continuous Delivery”

And how it further improves CI

*The “**Delivery**” process describes a complete project lifecycle end-to-end.*

“


CI
Func. Test
Pre Production
Delivery


CI
Func. Test
Pre Production
Continuous Delivery

The idea

Continuous delivery is a software engineering approach in which teams **produce** software **in short cycles**.

CI is not just about testing

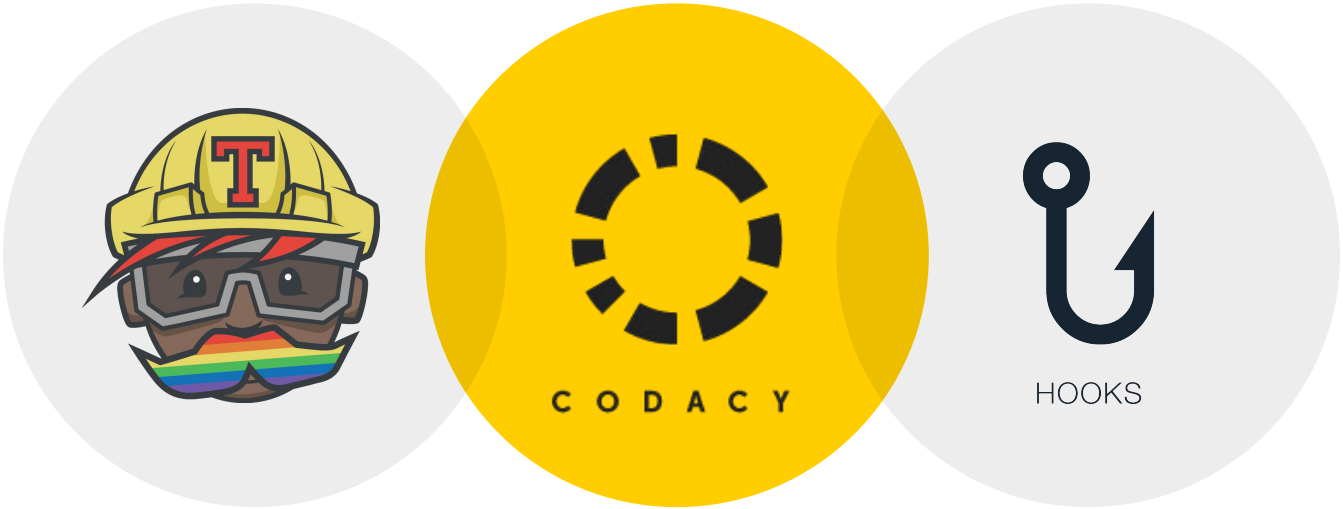
But also **improving** the quality of the **code** and making the life **easier** to the ***programmer.***

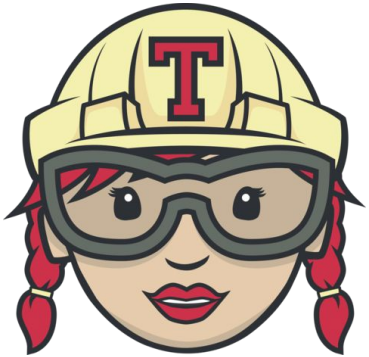
3

Configuring a project with **CI tools**, *from scratch*

Hands-on project

Tools used





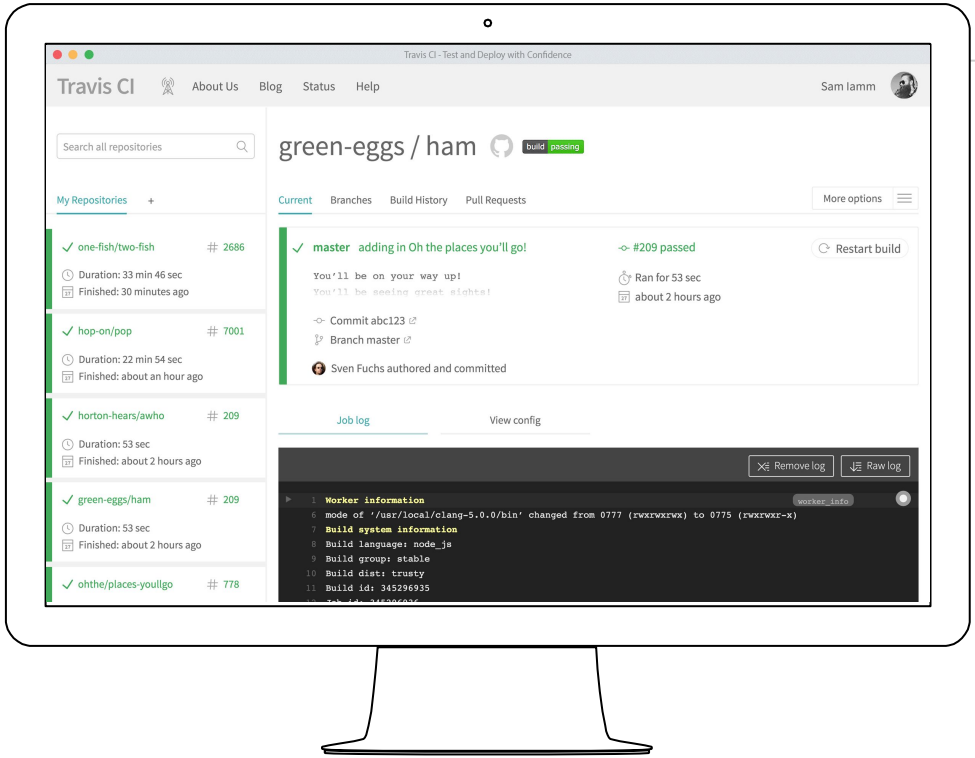
3.1

Travis CI

Continuous Integration and testing

Travis CI

- Automated test execution
- Get reports on your **email**
- Check code against different **versions of the language**



Configuring Travis CI

- Enable the repository on the **Travis CI web**
- Add a .travis.yml file to the repository
- Specify the **commands** used on the **integration** process

Important!

- **Travis CI** is only compatible with Java projects that use **Ant, Maven** or **Gradle**.

- No **out-of-the-box** support for **Eclipse.**

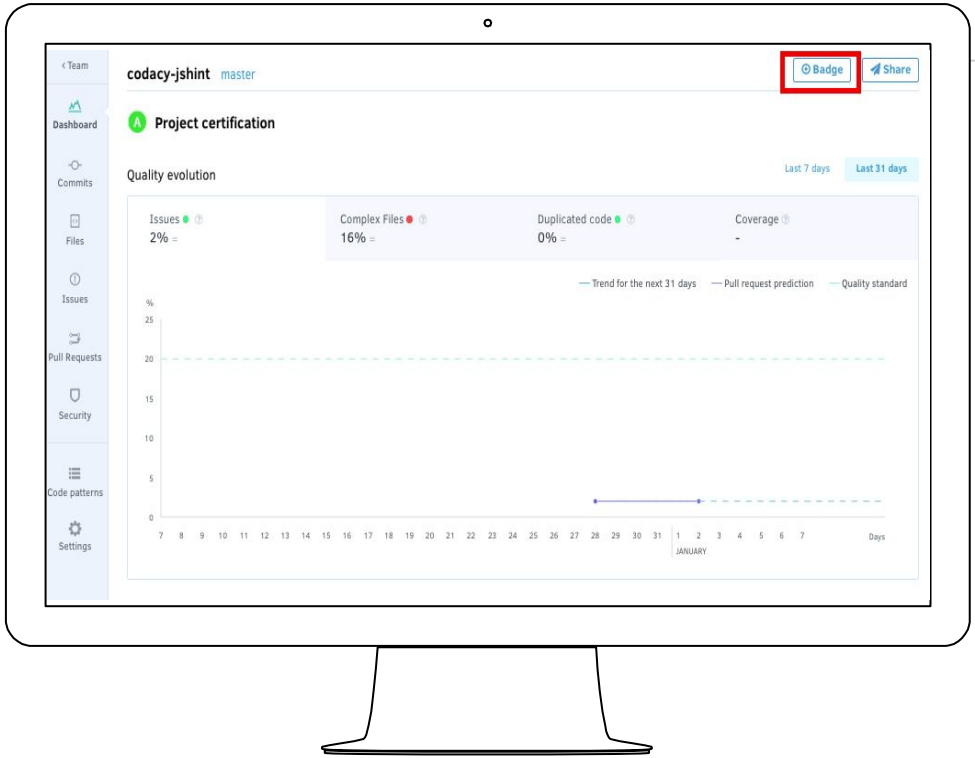
3.1

Codacy

Code quality analysis

Codacy

- Automated code quality report
- Get reports on the **evolution** of code quality
- Check code against popular **linters**



A **“Linter”** *is a tool that finds* ***syntax errors*** *or* ***style errors*** *on the code, returning a* ***report*** *to the user.*

Configuring Codacy

- Login on the Codacy web with your **Github account**
- Enable **Codacy** to access the repository

3.1

Git hooks

Pre-commit automated tasks

Git hooks

- **Automatic script execution** with git

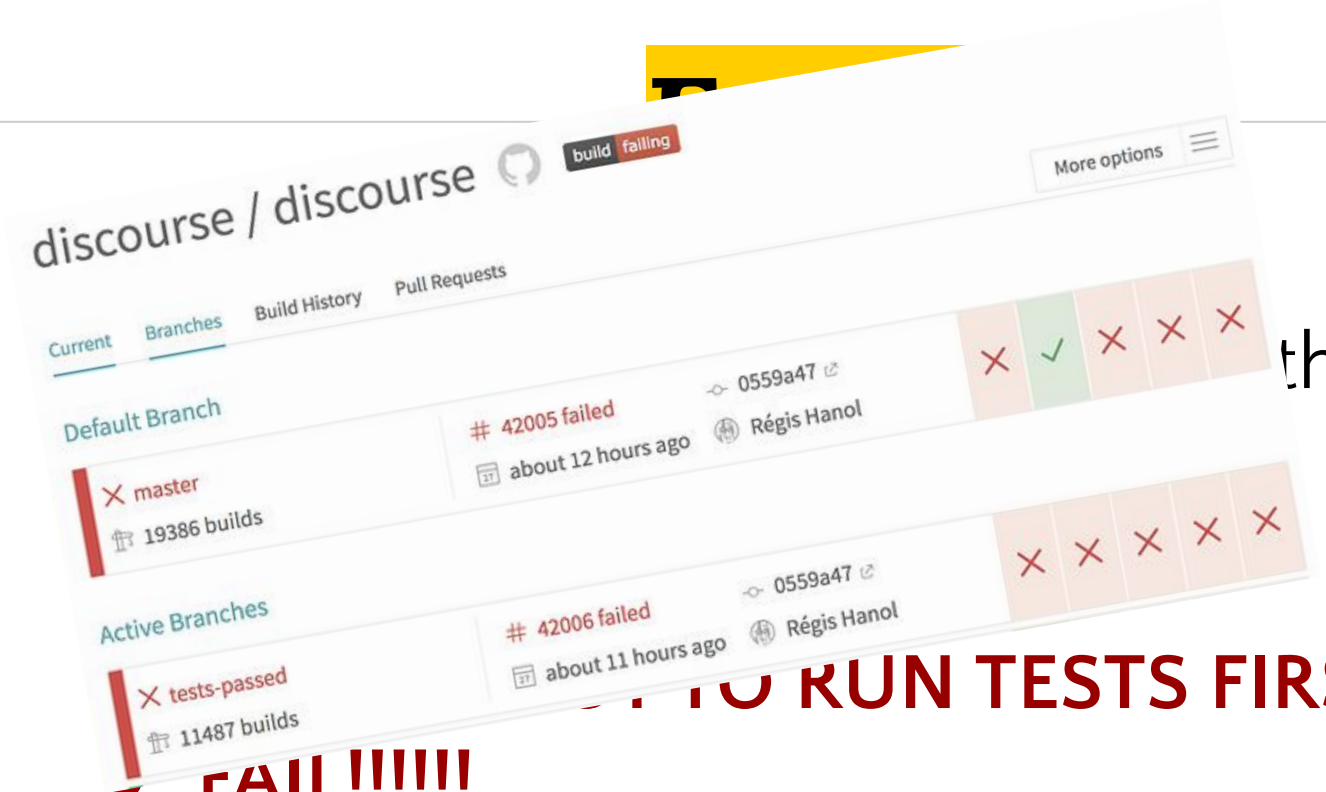
- Define **hooks** (scripts) that are executed before/after a git commit, pull, push, merge...

Example

- Let's suppose we did a **push** to the **team repository...**

- **BUT FORGOT TO RUN THE TESTS FIRST!**
- **FAIL!!!!!!**

oh no...



the team

TO RUN TESTS FIRST!

FAIL!!!!!!

WHAT I'VE DONE WRONG

RUN TESTS FIRST!

FAIL!!!!!!

OH NO

WHA

OH NO

Git hooks to the rescue

- We will configure a **git hook** that **runs tests** automatically **before a commit**.

- Won't let us push **unless all tests are passing.**

Configuring Git hooks

- Create the **hook** and add it to the .git/hooks folder
- Add a **gradle task** to automatically **install** the hooks

Surprise exam!

kahoot.it

Thanks!

Any questions ?

Bibliography

- http://deg.egov.bg/LP/core.base_concepts/guidances/concepts/delivery_process_CFEBBC78.html
- https://en.wikipedia.org/wiki/Continuous_delivery
- https://es.wikipedia.org/wiki/Entrega_continua
- https://dzone.com/articles/what-is-continuous-delivery-pipeline
- https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/herramientas-de-integracion-continua
- https://es.wikipedia.org/wiki/Integraci%C3%B3n_continua
- https://medium.com/@ajamaica/que-es-la-integraci%C3%B3n-continua-y-por-que-debr%C3%ADas-usarla-aea591f2a7d9