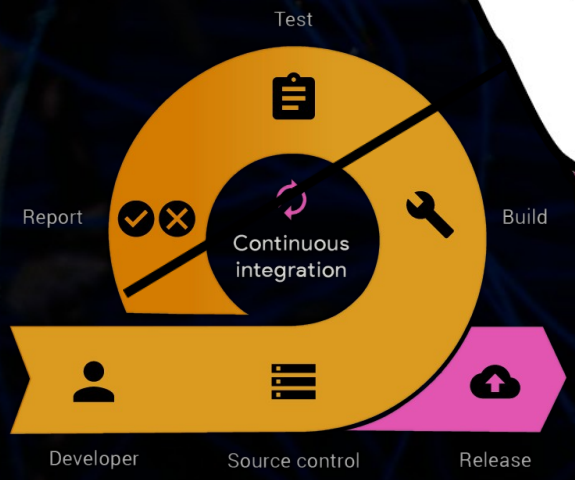
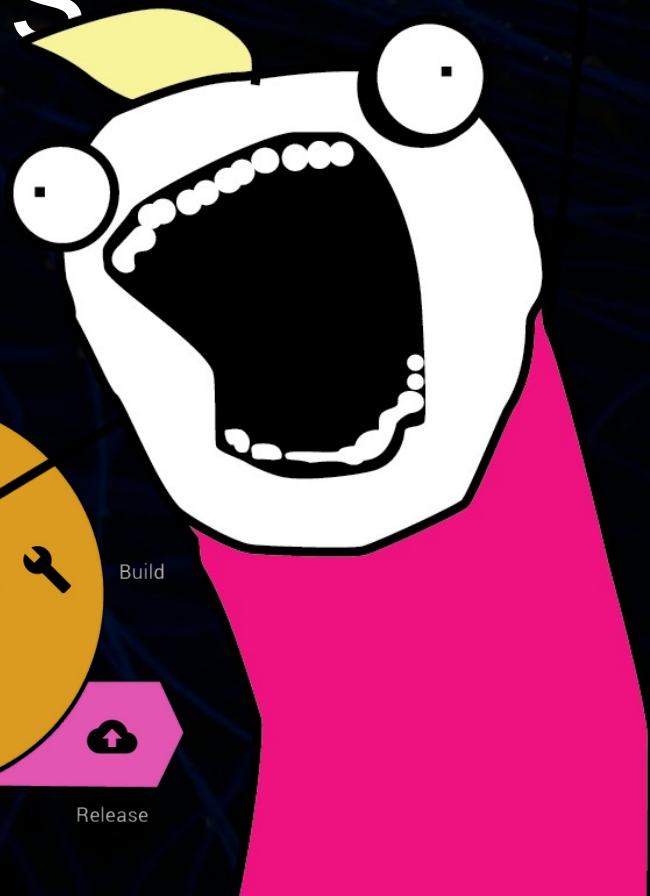
Continuous Integration All The Things

Anne Jan Brouwer

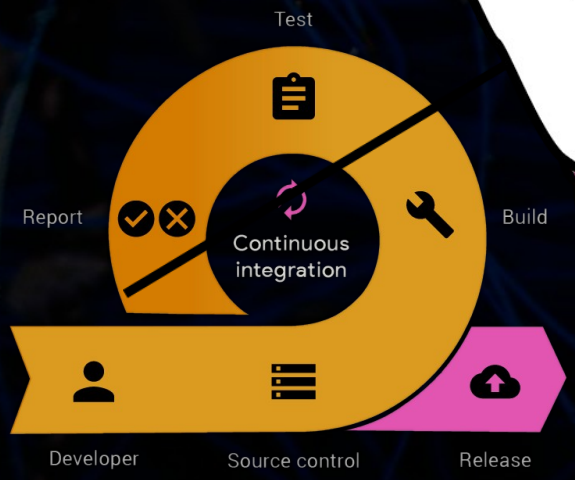


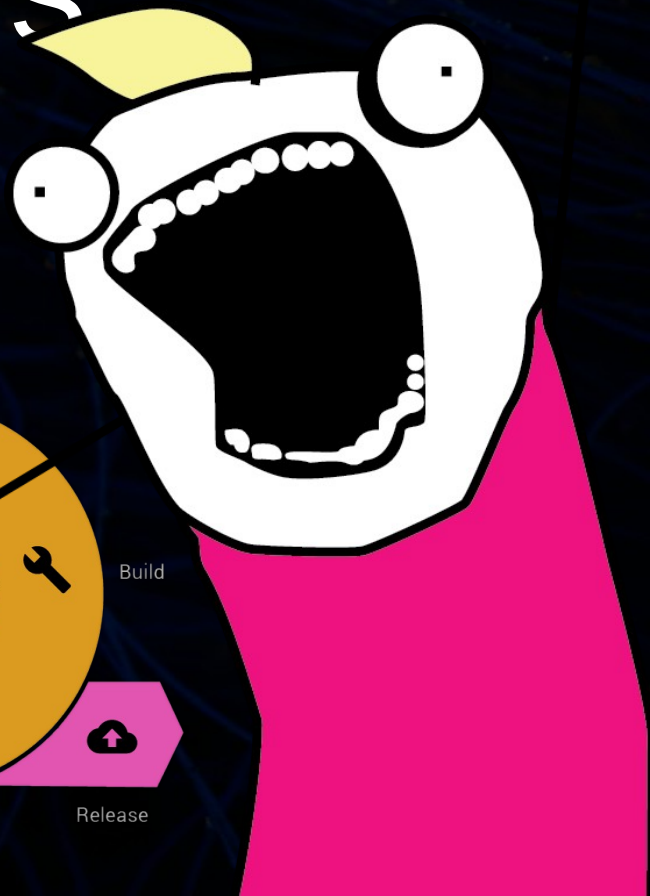


Continuous Integration All The Things

Anne Jan Brouwer

HACKER HOTEL





Anne Jan Brouwer

@annejanbrouwer

github.com/annejan

Developer / Hacker / Demoscener

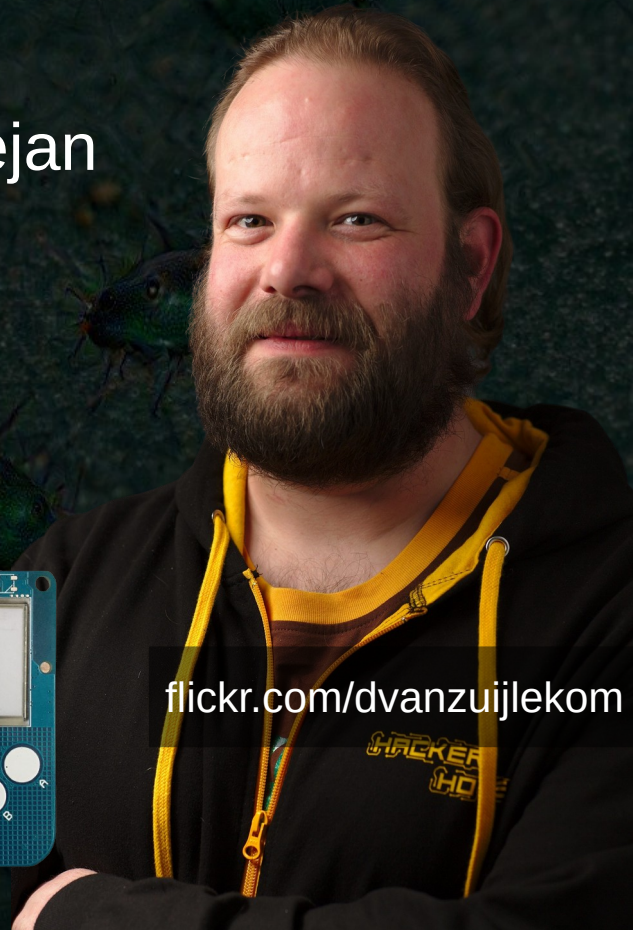
NOPROTOCOL://

de FEEST

Badge.team @ sha2017 / mch202

QtPass.org

Scene.rs



flickr.com/dvanzuijlekom

What is CI

Continuous integration is the practice of merging all developers' working copies to a shared mainline several times a day

Grady Booch 1991

Did not say “several times a day”

What is CI really

Reducing regression by making sure changes don’t accidentally break existing features and API contracts

Anne Jan Brouwer 2017

During a lunch conversation
about development cost

What is CI really

Continuous Improvement

Anne Jan Brouwer 2022

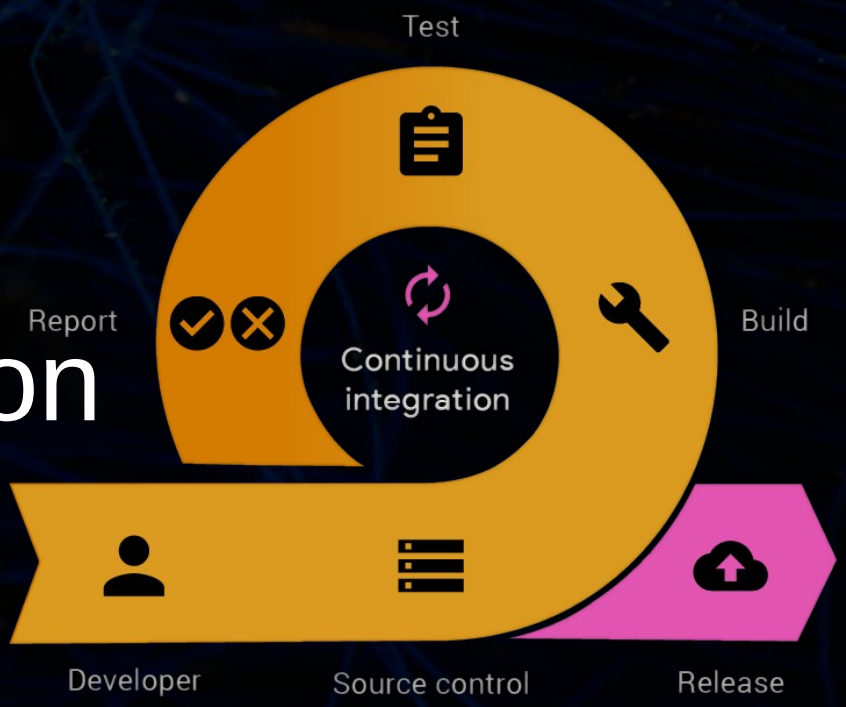
During a discussion on why doing fixes should be test driven

Why CI

Easier co-working

Preventing regression

Proving code works



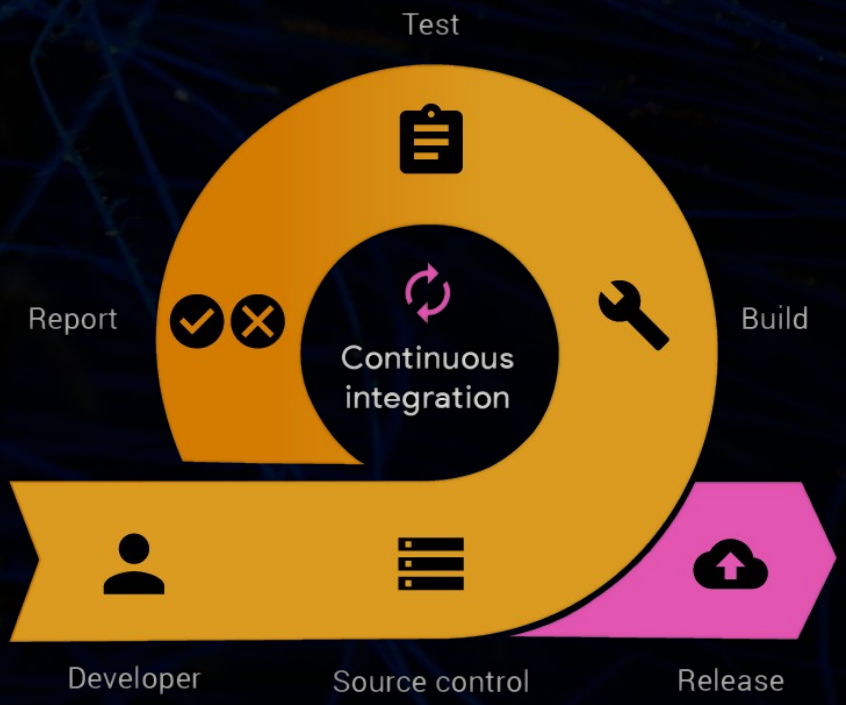
How to CI

Have a repository

Have tests

Automate builds

Automate testing



Create tests for bugfixes

pagerduty.com

Some common CI platforms

- GitLab CI
- Github Actions
- Jenkins
- Travis
- CircleCI
- Appveyor
- Buildkite

Some common CI platforms

- GitLab CI
- Github Actions
- Jenkins
- ~~Travis~~
- CircleCI
- Appveyor
- Buildkite

Basic example

```
mkdir my-ci-example && cd my-ci-example

echo -e "# I like CI\nBecause it helps" > file.md

git init && git add . && git commit -m "First commit"
git remote add origin git@github.com:annejan/mce.git
git push -u origin main
```

I like CI

Because it helps

Basic example

```
mkdir -p .github/workflows
cat << EOF > .github/workflows/md.yml
name: lint Markdown file
on: [push, pull_request]
jobs:
  lint:
    runs-on: ubuntu-latest
    steps:
    - uses: actions/checkout@v2
    - uses: actionshub/markdownlint@main
EOF

git add . && git commit -am "2nd commit" && git push
```

```
name: lint Markdown file
on: [push, pull_request]
jobs:
  lint:
    runs-on: ubuntu-latest
    steps:
    - uses: actions/checkout@v2
    - uses: actionshub/markdownlint@main
```

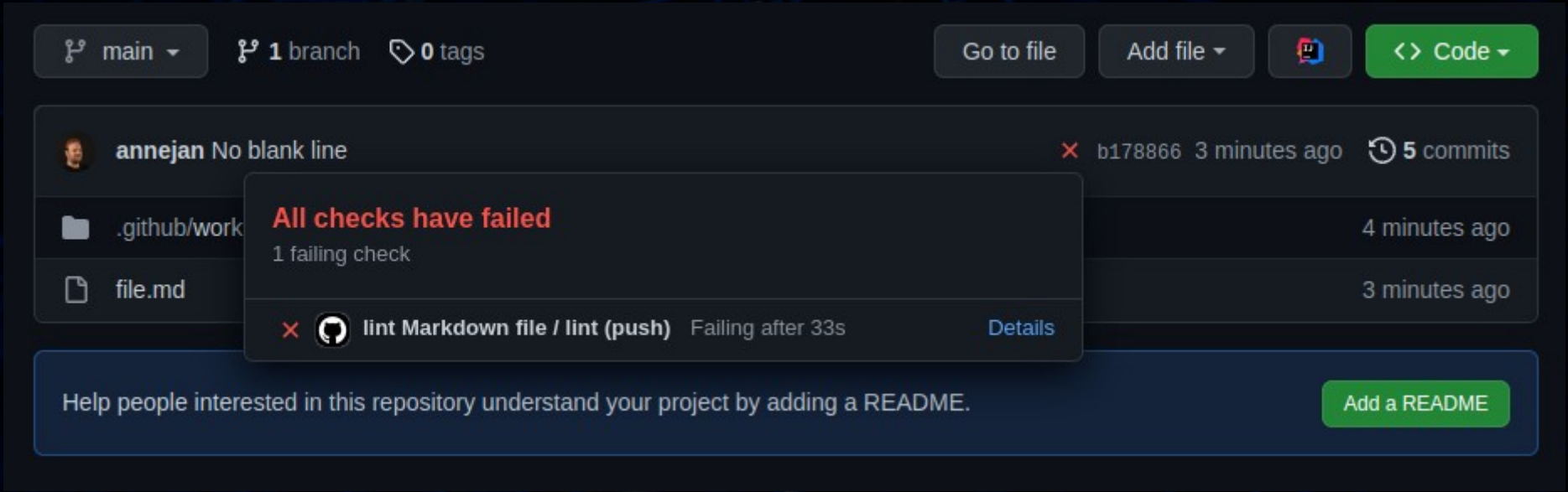
Realistic example

```
mkdir -p .github/workflows

cp ~/templates/md.yml .github/workflows/

git add . && git commit -m "2nd commit" && git push
```

Linting failed



Linting failed

lint

failed yesterday in 16s

Search logs

- Set up job 1s
- Build actionshub/markdownlint@main 13s
- Run actions/checkout@v2 0s
- Run actionshub/markdownlint@main 0s

```
1  ▶ Run actionshub/markdownlint@main
4  /usr/bin/docker run --name f70fbc7cf90c479440d93f0964ca45a4722_4dd1df --label 817f70 --workdir /github/workspace --rm -e "INPUT_PATH" -e "INPUT_FILESTOIGNOREREGEX" -e "HOME" -e "GITHUB_JOB" -e
   "GITHUB_REF" -e "GITHUB_SHA" -e "GITHUB_REPOSITORY" -e "GITHUB_REPOSITORY_OWNER" -e "GITHUB_RUN_ID" -e "GITHUB_RUN_NUMBER" -e "GITHUB_RETENTION_DAYS" -e "GITHUB_RUN_ATTEMPT" -e "GITHUB_ACTOR" -e
   "GITHUB_TRIGGERING_ACTOR" -e "GITHUB_WORKFLOW" -e "GITHUB_HEAD_REF" -e "GITHUB_BASE_REF" -e "GITHUB_EVENT_NAME" -e "GITHUB_SERVER_URL" -e "GITHUB_API_URL" -e "GITHUB_GRAPHQL_URL" -e
   "GITHUB_REF_NAME" -e "GITHUB_REF_PROTECTED" -e "GITHUB_REF_TYPE" -e "GITHUB_WORKSPACE" -e "GITHUB_ACTION" -e "GITHUB_EVENT_PATH" -e "GITHUB_ACTION_REPOSITORY" -e "GITHUB_ACTION_REF" -e "GITHUB_PATH"
   -e "GITHUB_ENV" -e "GITHUB_STEP_SUMMARY" -e "RUNNER_OS" -e "RUNNER_ARCH" -e "RUNNER_NAME" -e "RUNNER_TOOL_CACHE" -e "RUNNER_TEMP" -e "RUNNER_WORKSPACE" -e "ACTIONS_RUNTIME_URL" -e
   "ACTIONS_RUNTIME_TOKEN" -e "ACTIONS_CACHE_URL" -e GITHUB_ACTIONS=true -e CI=true -v "/var/run/docker.sock":"/var/run/docker.sock" -v "/home/runner/work/_temp/_github_home":"/github/home" -v
   "/home/runner/work/_temp/_github_workflow":"/github/workflow" -v "/home/runner/work/_temp/_runner_file_commands":"/github/file_commands" -v "/home/runner/work/my-ci-example/my-ci-
   example":"/github/workspace" 817f70:fbc7cf90c479440d93f0964ca45a4722
5  Error: .//file.md:1: MD022 Headers should be surrounded by blank lines
6
7  A detailed description of the rules is available at https://github.com/markdownlint/markdownlint/blob/master/docs/RULES.md
```

- Post Run actions/checkout@v2 0s
- Complete job 0s

Fixing the Markdown

```
sed -i '1 a \\' file.md

mdl file.md

git status
git commit -am "Fixed Markdown header whitespace"

git push
```

```
diff --git a/file.md b/file.md
index 0b0ad1d..e74dcd4 100644
--- a/file.md
+++ b/file.md
@@ -1,2 +1,3 @@
# I like CI
+
Because it helps
```

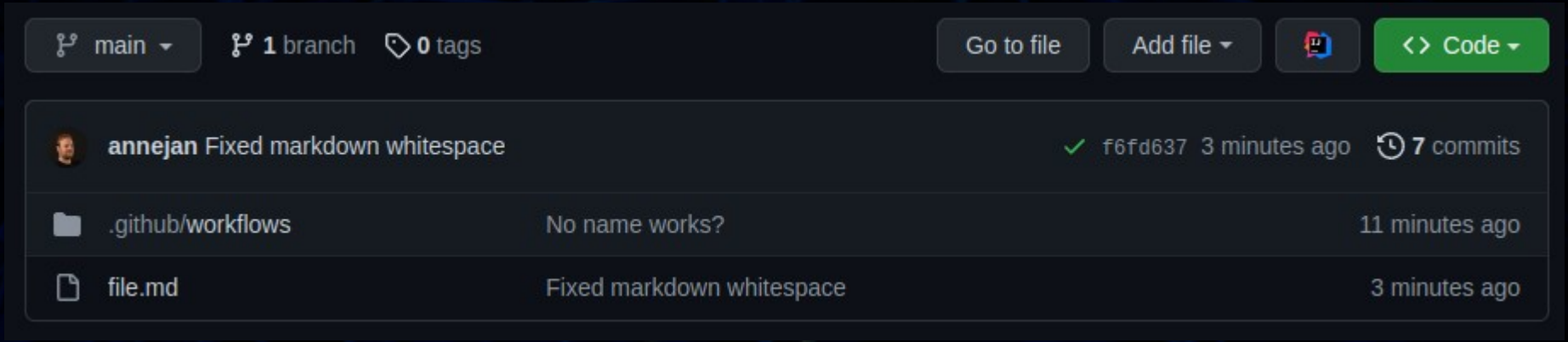
I like CI

Because it helps

Linting failed

Error: **.//file.md:1: MD022 Headers should be surrounded by blank lines**

Linting succeeded



Linting succeeded

Fixed markdown whitespace lint Markdown file #4

Re-run all jobs

Summary

Jobs

lint

lint

succeeded 2 minutes ago in 28s

Search logs

Set up job	1s
Build actionshub/markdownlint@main	25s
Run actions/checkout@v2	1s
Run actionshub/markdownlint@main	1s
Post Run actions/checkout@v2	0s
Complete job	0s

What to CI

If you can lint it

If you can analyze it

If you can build it

If you can run it

If you can test it

Levels of CI

Syntax and coding standards

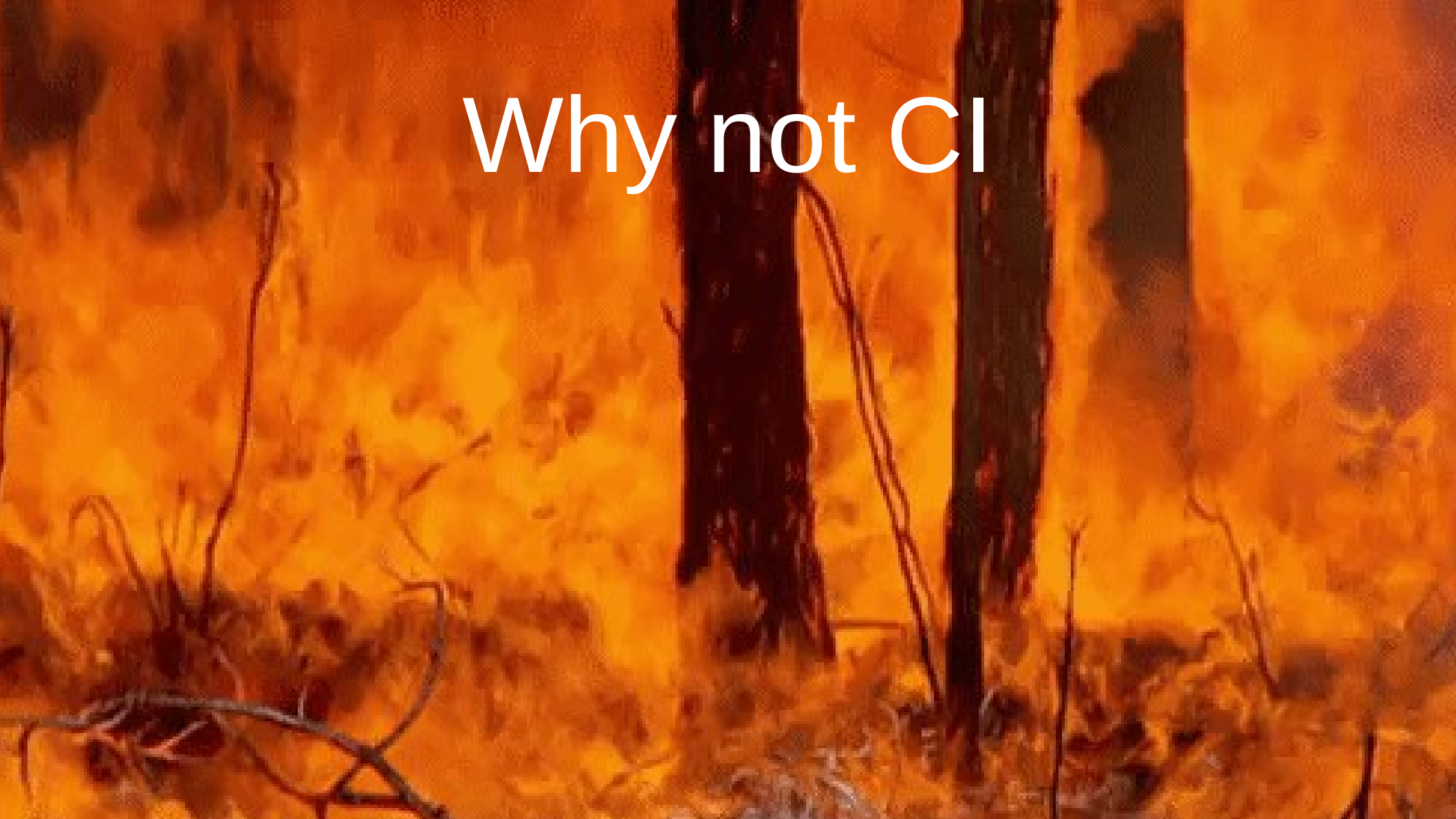
Code flow and data types

Dependency availability & compilation

Code execution and basic functioning

Unit / Feature / End tot End testing

Why not CI



Real world examples

QtPass

I don’t run windows, how do I make QtPass-installer.exe

Code quality metrics

Some years ago

Travis

- Free for OSS \o/
  - Was only temporary : (

AppVeyor

- Still builds the Windows installer

Now

Github Super-Linter

Needs action(s) for Mac and Linux

- Just do it™

Super-Linter

Lints a lot of languages and file types!

> 42 types!!

https://github.com/github/super-linter

Super-Linter

Works great for some languages

Including English

Used and supported by GitHub

Under active development

Now

~~Github Super-Linter~~

Auto generated documentation

- 340 page whitepaper

Needs action(s) for Mac and Linux

Now

Seriously looking for volunteers

help@qtpass.org

hatchery.badge.team

Code Coverage is a fun game

build	passing
maintainability	A
test coverage	94%
code quality	A
codecov	97%
license scan	passing
vulnerabilities	0

hatchery.badge.team

Code Coverage is an addicting game

Code coverage report for **All files**

Statements: **83.82%** (290 / 346) Branches: **52.24%** (70 / 134) Functions: **77.63%** (59 / 76) Lines: **87.8%** (288 / 328) Ignored: none

File	Statements		Branches		Functions		Lines	
server/	88.24%	(30 / 34)	62.50%	(5 / 8)	83.33%	(5 / 6)	90.63%	(29 / 32)
server/api/auth/	87.50%	(21 / 24)	83.33%	(10 / 12)	80.00%	(4 / 5)	91.30%	(21 / 23)
server/api/screenshot/	84.00%	(63 / 75)	61.54%	(16 / 26)	88.89%	(16 / 18)	91.18%	(62 / 68)
server/api/user/	87.18%	(68 / 78)	61.54%	(16 / 26)	80.95%	(17 / 21)	90.67%	(68 / 75)
server/config/	78.72%	(74 / 94)	30.00%	(6 / 20)	57.89%	(11 / 19)	81.32%	(74 / 91)
server/config/environment/	68.75%	(11 / 16)	39.47%	(15 / 38)	0.00%	(0 / 1)	68.75%	(11 / 16)
server/db/	100.00%	(8 / 8)	100.00%	(0 / 0)	100.00%	(1 / 1)	100.00%	(8 / 8)
server/screenshotCapture/	88.24%	(15 / 17)	50.00%	(2 / 4)	100.00%	(5 / 5)	100.00%	(15 / 15)

hatchery.badge.team

Code Coverage is a fun game

Swagger / OpenAPI

hatchery.badge.team

https://hatchery.badge.team/api

Swagger
Supported by SMARTBEAR

https://hatchery.badge.team/docs/api-docs.json

Explore

Hatchery by badge.team 0.2 OAS3

https://hatchery.badge.team/docs/api-docs.json

Simple micropython software repository for Badges.

Hatchery - Website
Send email to Hatchery

MIT

Basket Related to getting Projects for specific Badge models.

GET /basket/{badge}/list/json Get the latest released versions for a badge model.

Parameters

Try it out

Name	Description
badge * required string($slug)	sha2017

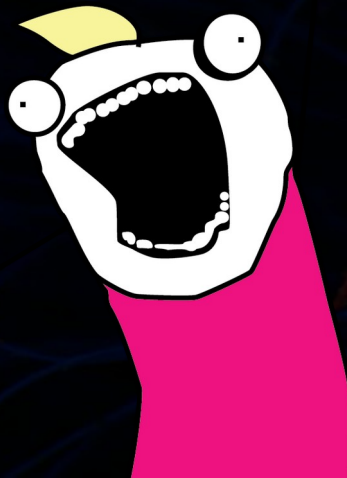
hatchery.badge.team

Code Coverage

Swagger / OpenAPI

~~Github Super-Linter~~

Just do it yourself



Super-Linter

- PHP support sucks!
- Probably other languages too ;-)

- Let’s do something about it

Super-Linter

- Let’s do something about it

When I find the time

Super-Linter

- PHP support sucks!
- Probably other languages too ;-)

- Let’s do something about it

Super-Linter

- PHP support sucks!
- Probably other languages too ;-)

- Let’s do something about it

hatchery.badge.team

Not just Hatchery

Needs more time and effort :-)

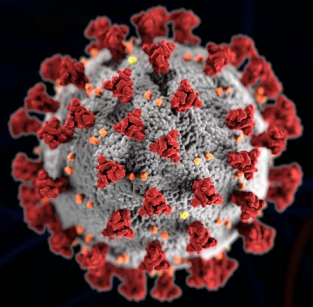
Firmware builds

Real world examples

MinVWS

- Exception routes
  - So many exceptions

VLA Projects

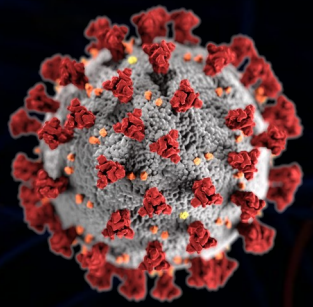


Real world examples

MinVWS

- Exception routes
  - So many exceptions
- End-to-end testing
- Chain testing
- “Borgen” or in English, API Guarantees

VLA Projects

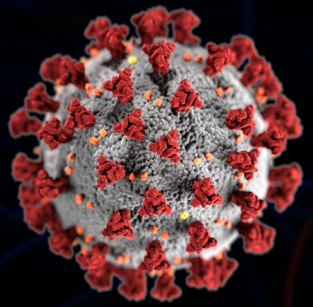


Real world examples

MinVWS

- Linting
- Static Analysis
- Unit / Feature testing
- End-to-end

VLA Projects



JSON Schema’s

MinVWS VLA Projects

Like OpenAPI / Swagger

Can be used for “everything”

Validation made easy

Real world examples

MinVWS VLA Projects

Lots of end-to-end testing

Cucumber language

Python

What to CI

If you can lint it

If you can analyze it

If you can build it

If you can run it

If you can test it

Questions?

Example questions:

- Which CI should I use?
- So what about CD?
- What about cost?
- Should I . . CI ?