

# Pipeline ETA: 3... 2... Tomorrow?

Let's Fix That!

Alexander Ptakhin, Tech Lead @ Prestatech / Berlin, DevOpsDays Zurich 2025

Context, always context

**Slow software is not the best  
example for us**

# Principles

- Make things simpler
- Make a sustainable fast feedback loop of client feedback

# Agenda

- Writing and supporting pipelines
- Speeding up long regression tests
- Continuous Integration: making PRs flow smoothly

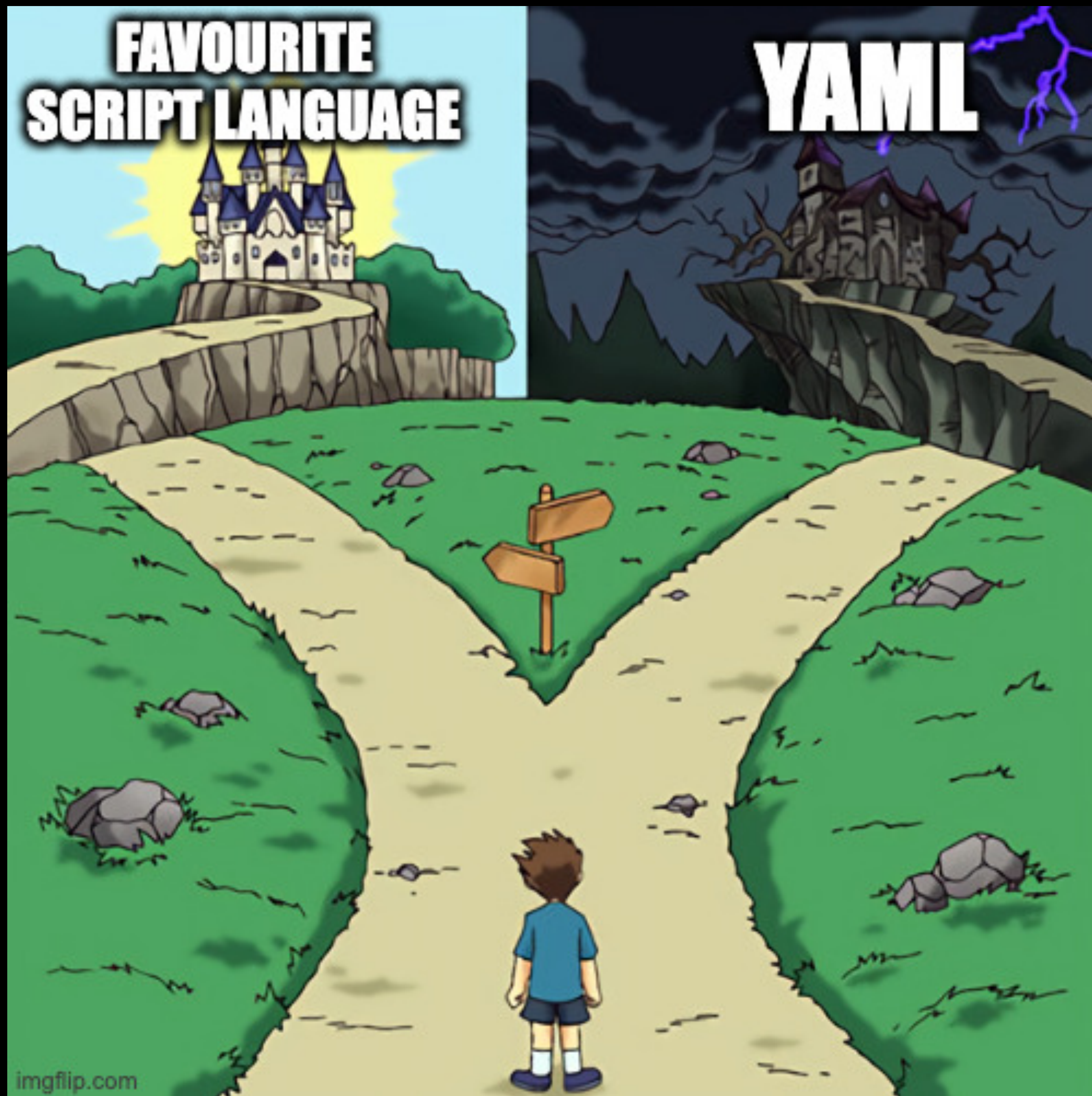
# 1. Writing and supporting pipelines

We write YAML in pipelines.

Who loves YAML in pipelines?

**FAVOURITE  
SCRIPT LANGUAGE**

**YAML**





# Runnable things outside

*Makefile:*

**fmt:**

`./shell/reformat-the-code.sh`

**line:**

`./shell/lint-the-code.sh`

Run:

`make fmt && make lint`



# Still in YAML

- Minimal setup for service connections in YAML (docker registries, k8s):
- Caching

*Build job:*

jobs:

  build:

    steps:

    - uses: actions/checkout@v4

    - # ... caching

    - run: make lint

    - run: make test

    - name: Build image

      run: make build tag=\${{ github.run\_id }}

    outputs:

      build\_id: \${{ github.run\_id }}

*Build job:*

jobs:

  build:

    steps:

    - uses: actions/checkout@v4

    - # ... caching

    - run: make lint

    - run: make test

    - name: Build image

      run: make build tag=\${{ github.run\_id }}

    outputs:

      build\_id: \${{ github.run\_id }}

*Deploy job:*

jobs:

  deploy:

    needs: build

    steps:

    - # ... connections

    - name: Deploy dev

      run: make deploy e=dev tag=\${{ needs.build.outputs.build\_id }}

## 2. Speeding up long regression tests

- Longer pipelines  $\Rightarrow$ 
  - Bigger batches of code changes  $\Rightarrow$ 
    - Longer feedback loop
    - Bigger pull request to review

# Speeding up long regression tests

- Test only things that give us quality
- Speed up the code

# Speeding up long regression tests

- Test only things that give us quality
- Speed up the code
- Parallelize tests (1 -> 2 -> 4 -> 8 jobs)
- Trigger only tests affected by the change

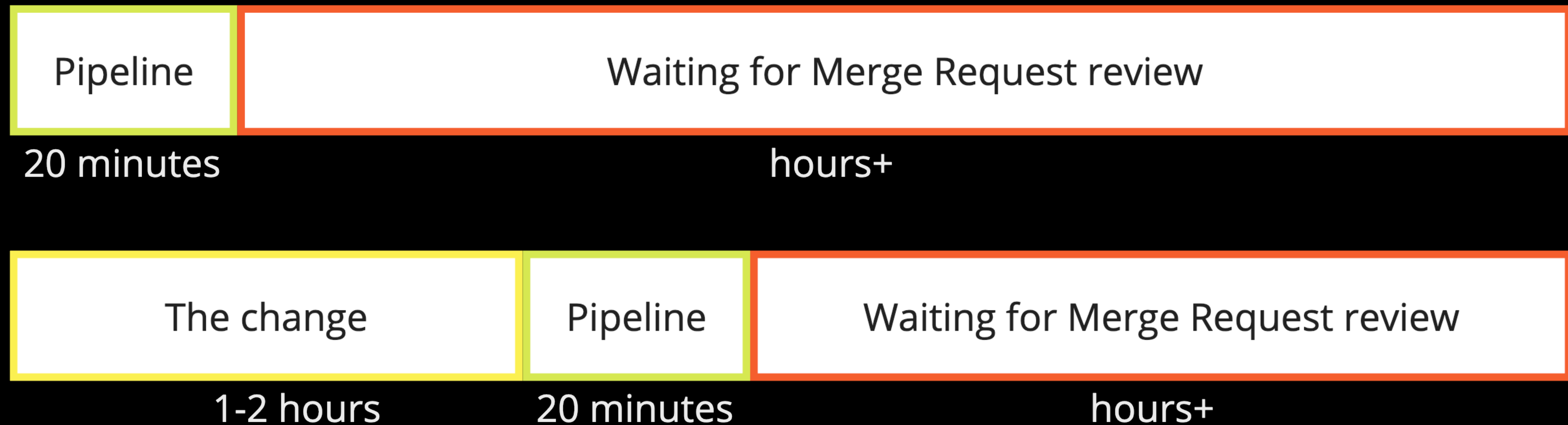
```
git diff --name-only origin/main  
Processing/Documents-Italy  
Processing/Documents-Germany
```

run only Documents-Italy and Documents-Germany tests, not e.g. Documents-UK



### 3. Continuous integration

- Continuous integration is the practice of integrating source code *changes frequently* while ensuring it's in a workable state



# Continuous integration

## Possible practices to try

- Async non-blocking post-commit review
- Sync code-review for bigger changes

# Continuous integration

## Possible practices to try

- Async non-blocking post-commit review
- Sync code-review for bigger changes
- Design review before first line of code
- Edge cases review with QA or team before first line of code

# Wrap-up: principles

- Make things simpler
  - Testable, runnable locally
- Make a sustainable fast feedback loop of client feedback
  - Local optimization may hit us. We need global ones

# Contacts

**Thank you for your attention!**

[SpeakerDeck.com](https://SpeakerDeck.com) presentation



Alexander Ptakhin contacts

