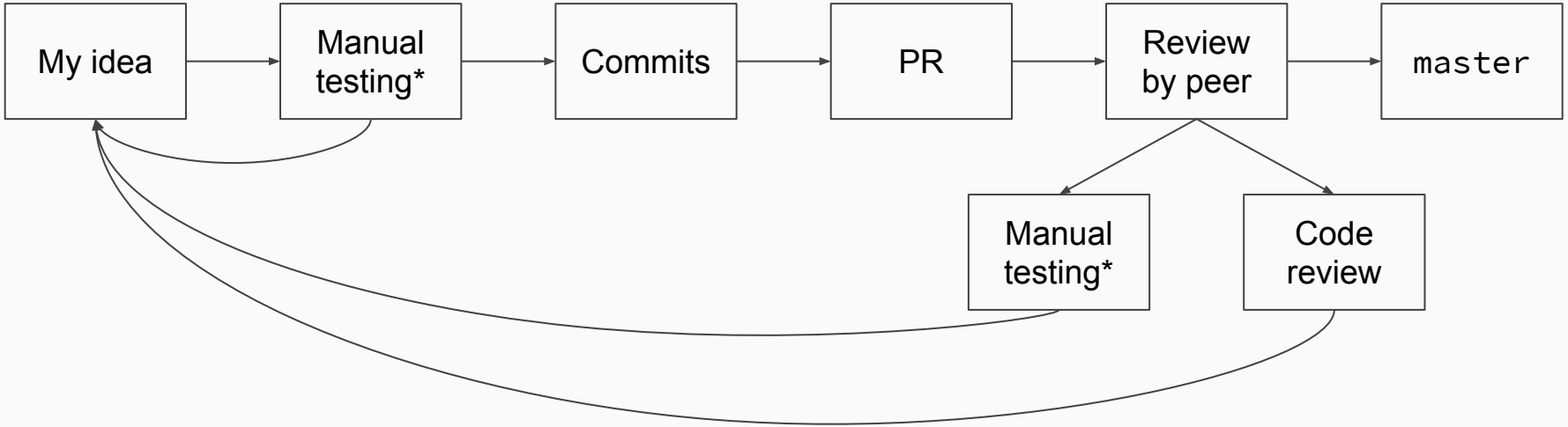
Continuous Integration

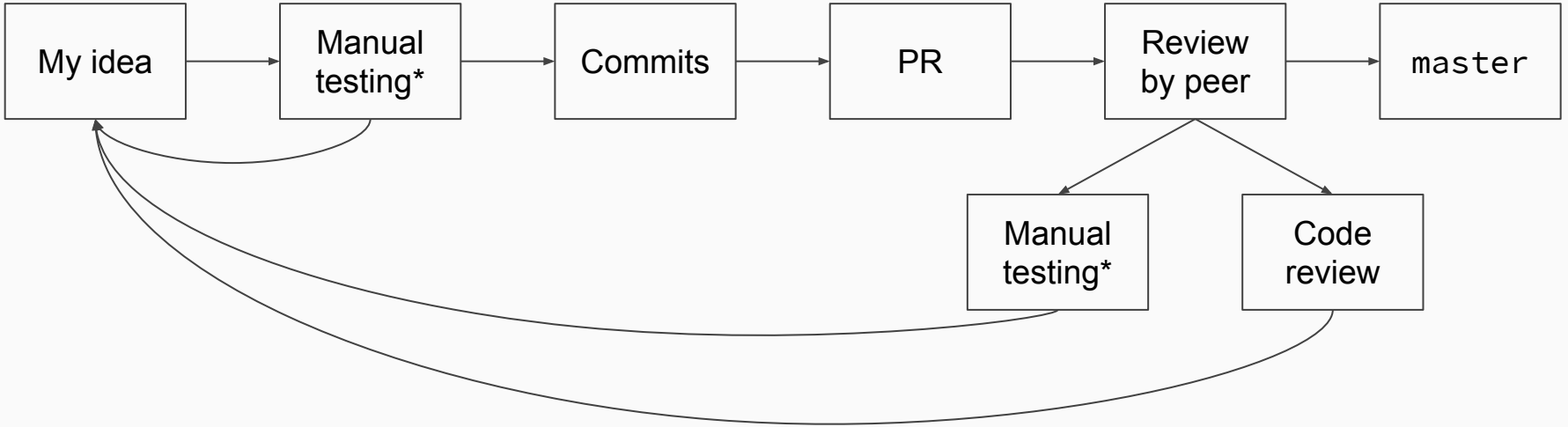
By Joren Vrancken

Manual workflow



*manually testing changes and manually running unit/integration/etc tests.

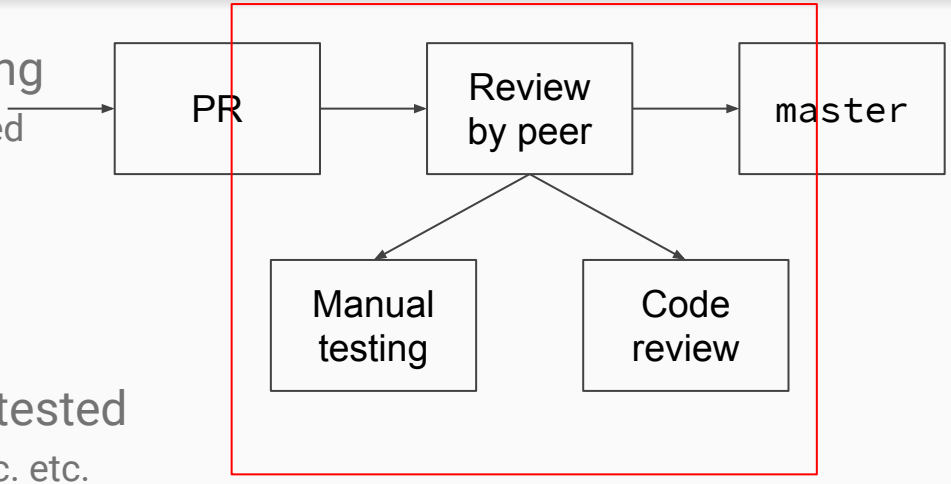
What Can Go Wrong?



*manually testing changes and manually running unit/integration/etc tests.

The human error in manual workflow

- Every change needs manual checking
  - Codestyle mistakes are easily overlooked
- No consistency in what is tested
  - Easy to forget something
  - Different views between reviewers
  - “This is so simple, I don’t have to test it”
- No consistency in how a change is tested
  - Different operating systems, laptops, etc. etc.



Continuous Integration

Automated tasks on your codebase that run under certain conditions.

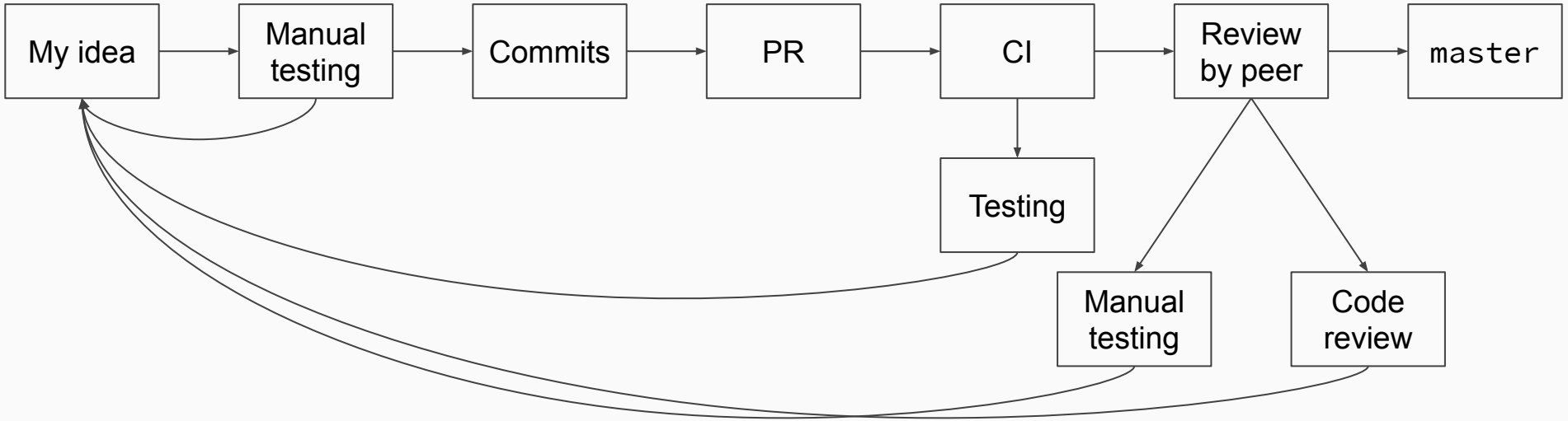
Goals of CI

- Automate running the boring/tedious/time consuming parts of a review
- Consistency in what is tested
- Consistency in test environment
  - Similar to the production environment

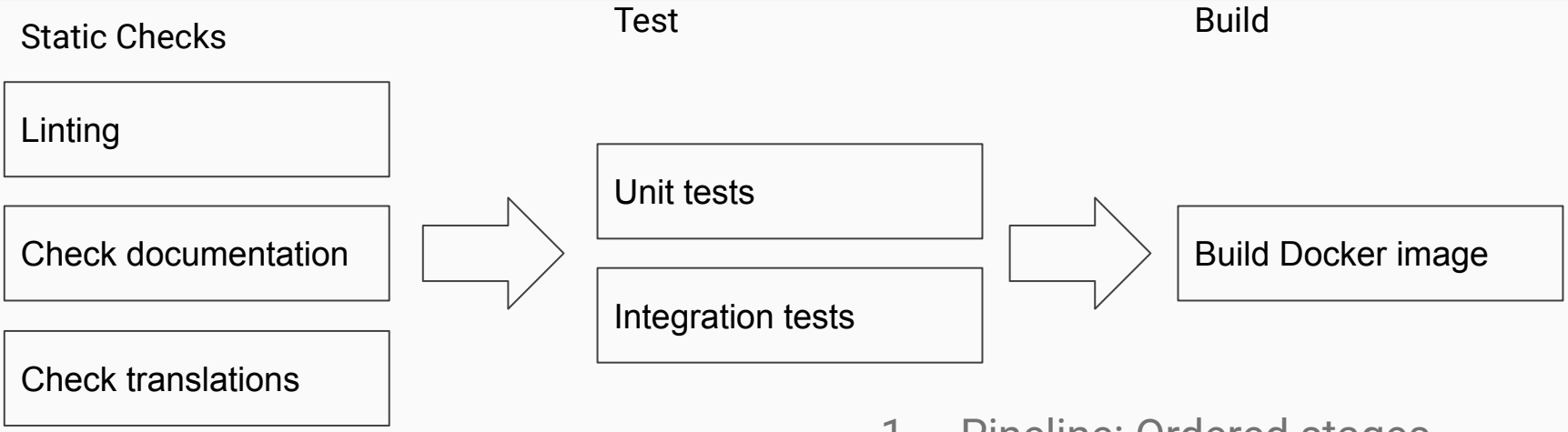
What to automate using CI?

- Running integration/unit tests
- Linting
- Checking whether documentation is up to date
- Checking for visual changes
- Whatever you want

Continuous Integration Workflow



CI Pipeline



1. Pineline: Ordered stages
2. Stage: One or more jobs
3. Jobs: Ordered steps to reach a goal

Continuous Deployment

- Deploy software using CI
- Automate building
- Automate deployment
  - Under specific conditions (e.g. only on the `master` branch)

GitHub Actions

- GitHubs own built-in CI platform
- Released last November and perceived as a development “game changer”
- More than CI
  - Distributable
  - Define steps using Unix commands, Javascript or Docker images
  - Execute jobs on events (e.g. starring, first contributor)

My first GitHub Actions Workflow

Demonstration

GitHub Actions Documentation

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