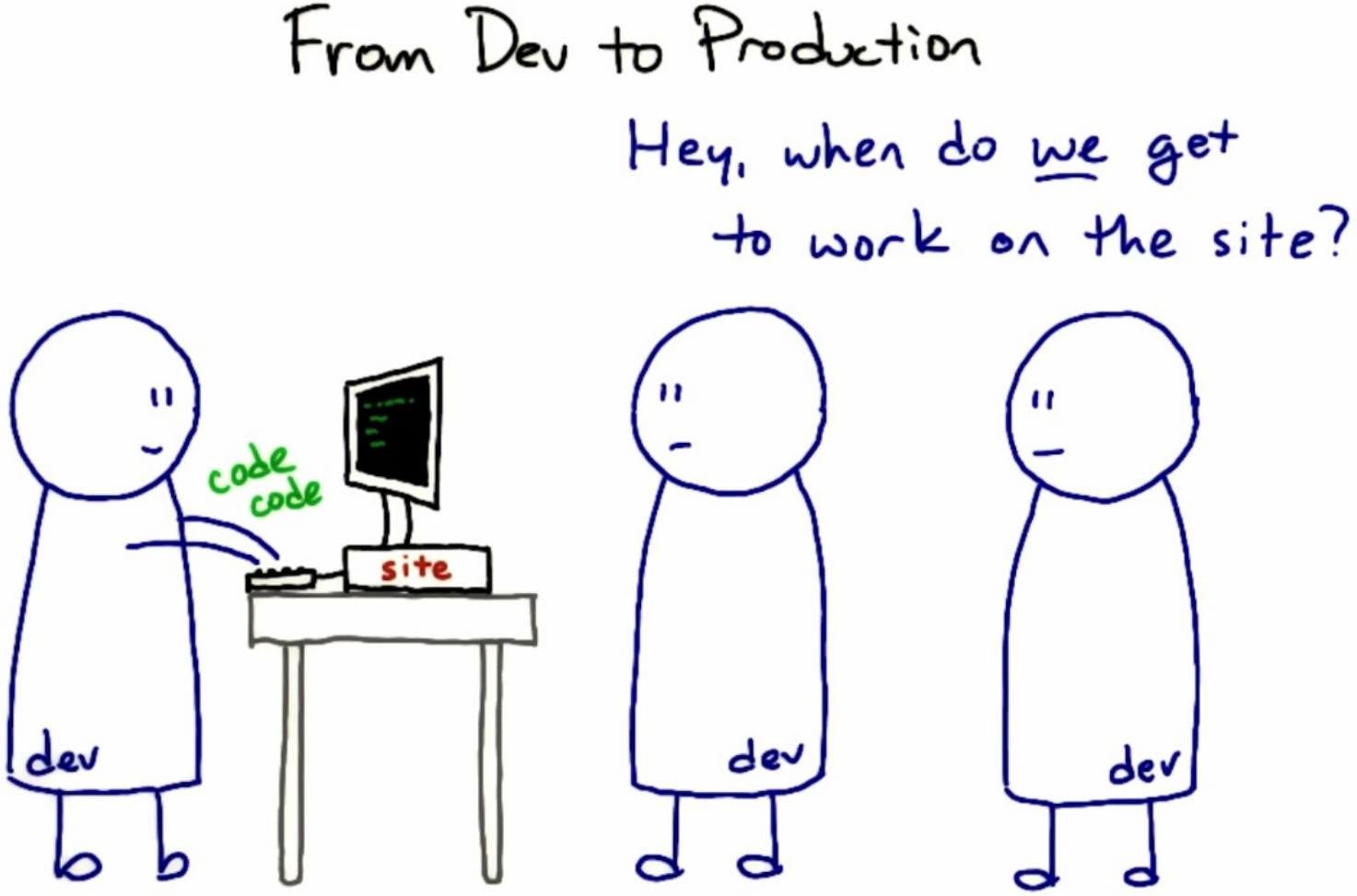
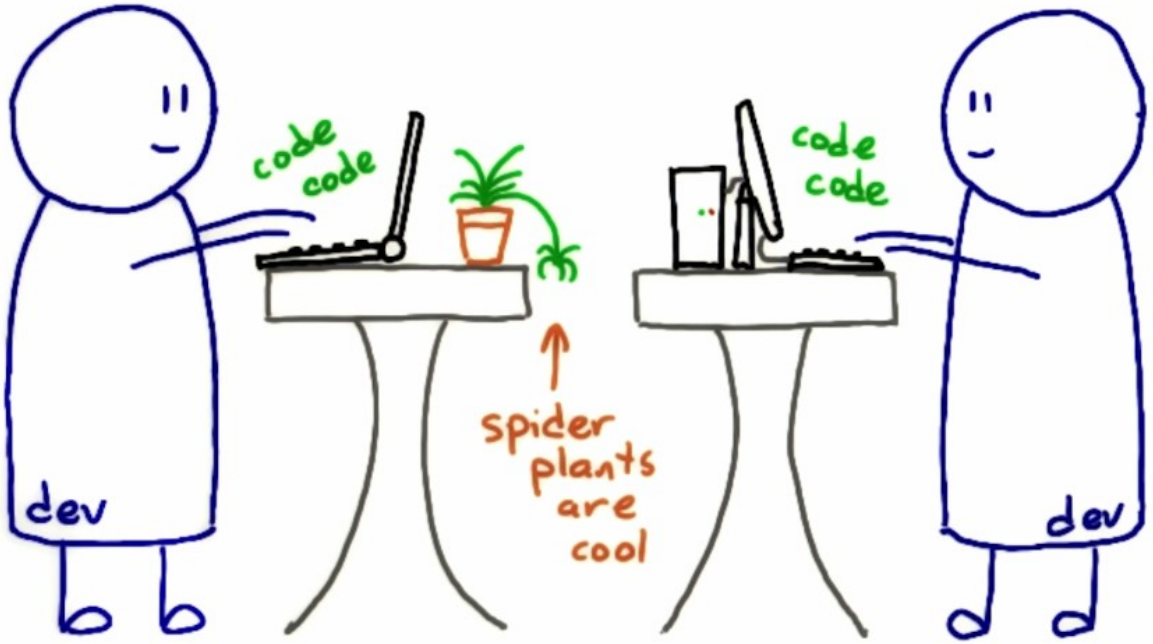
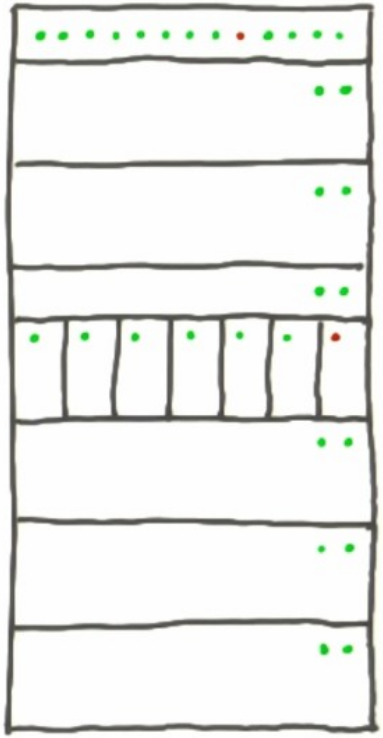
From Dev to Production

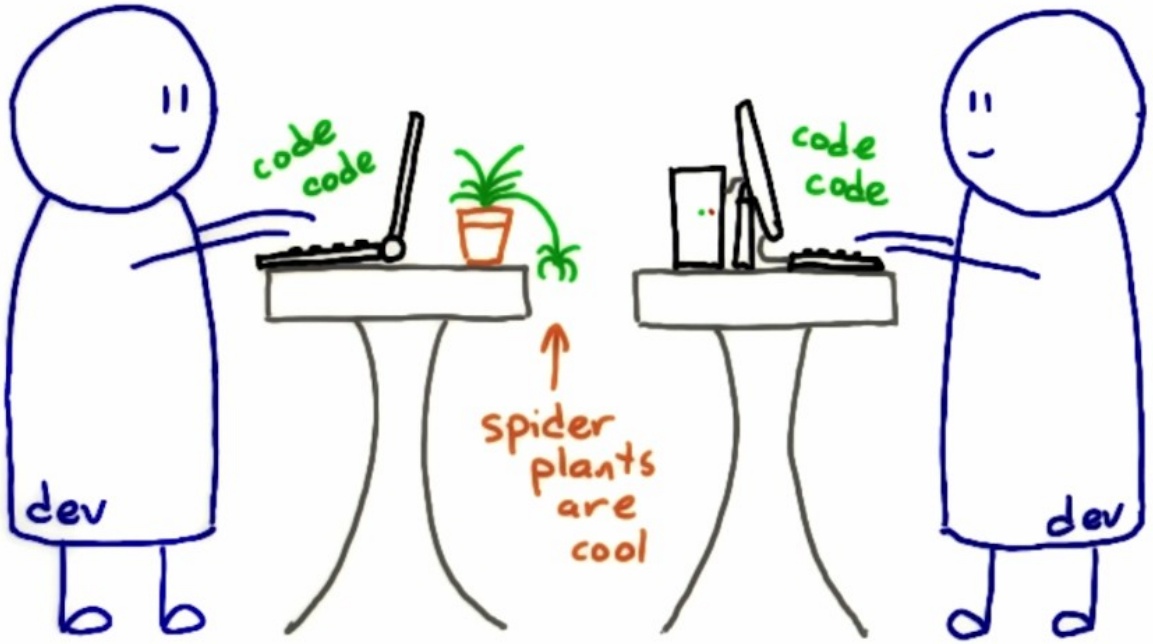


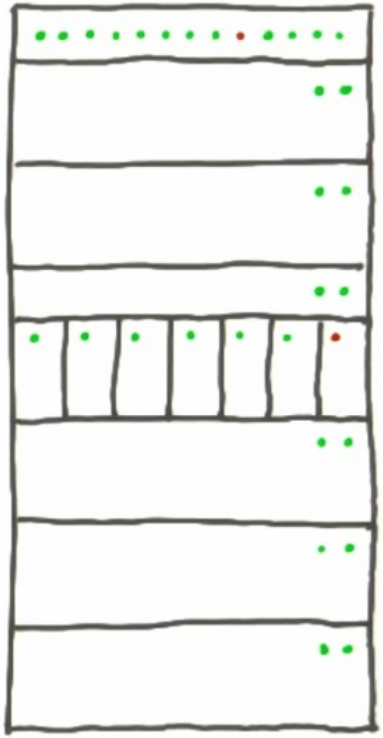
From Dev to Production



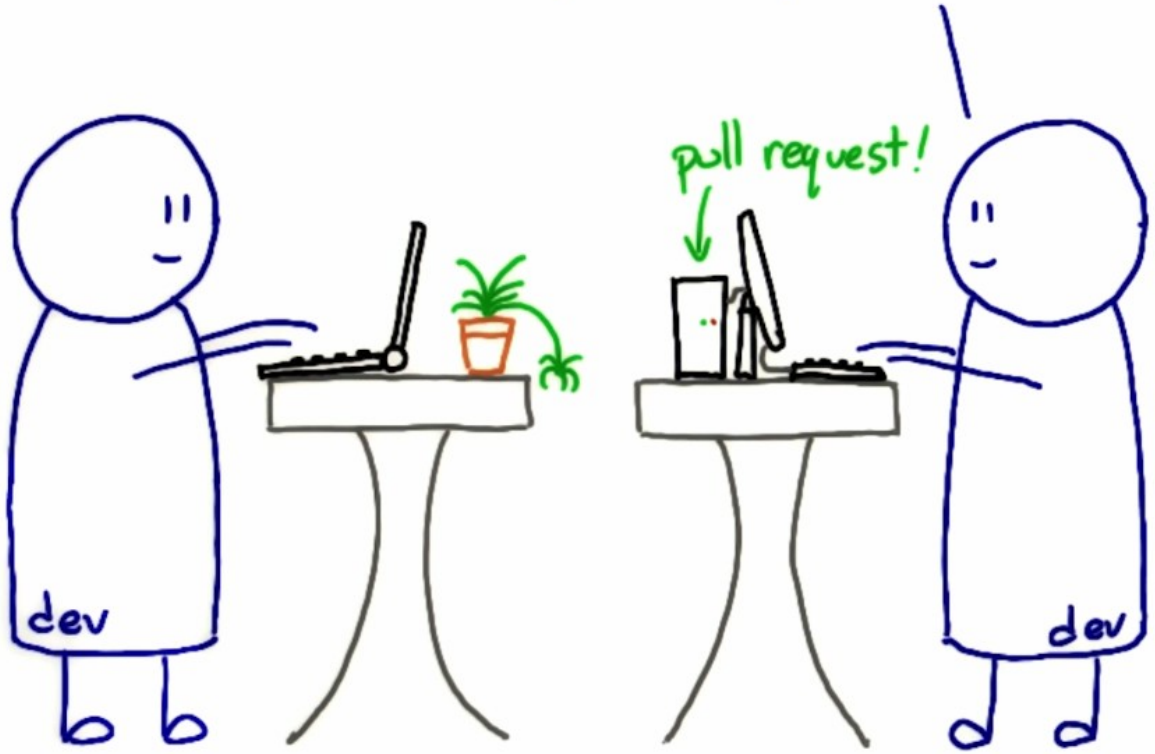


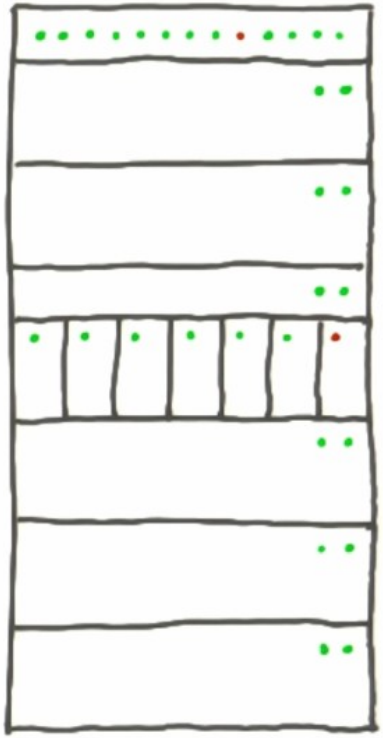
From Dev to Production



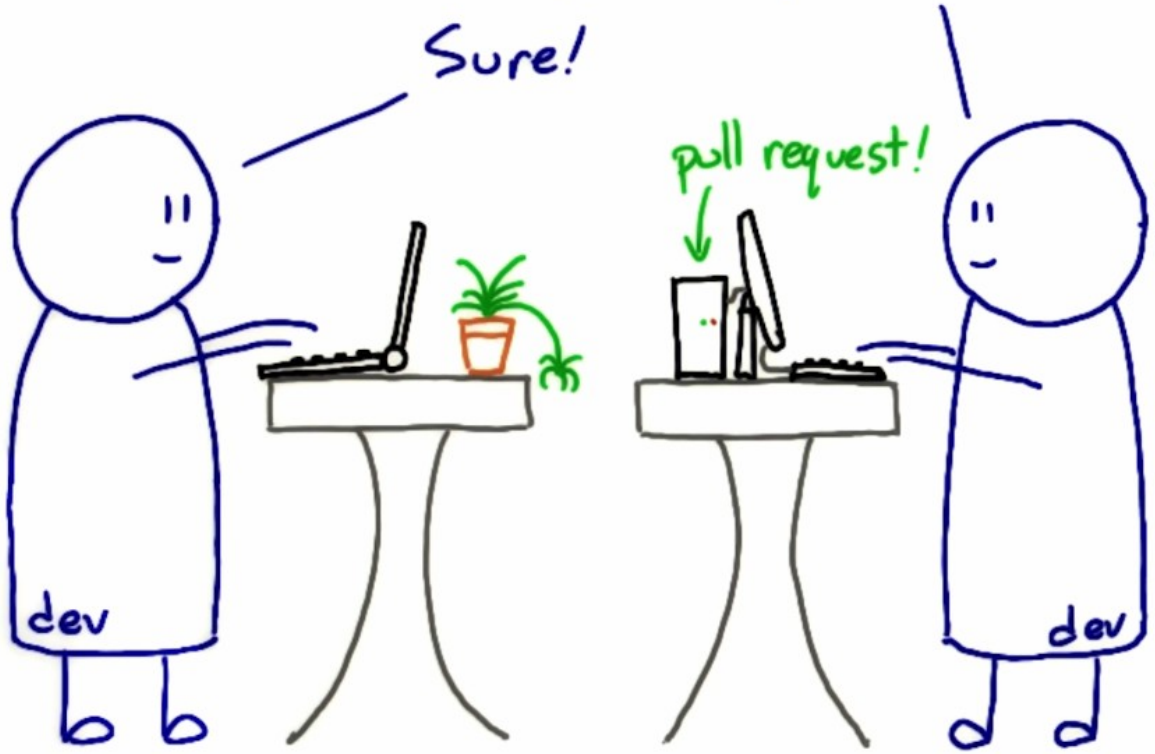


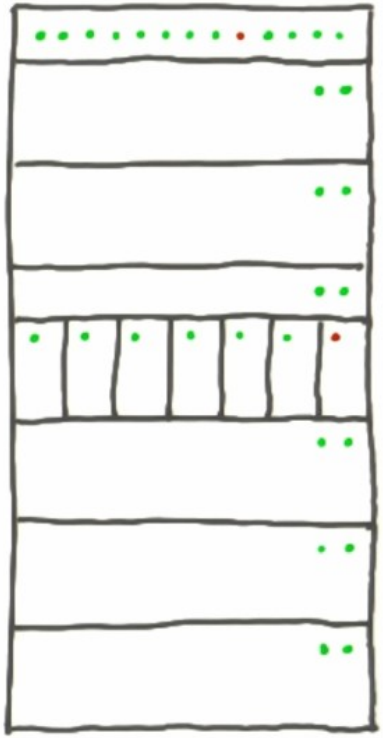
From Dev to Production



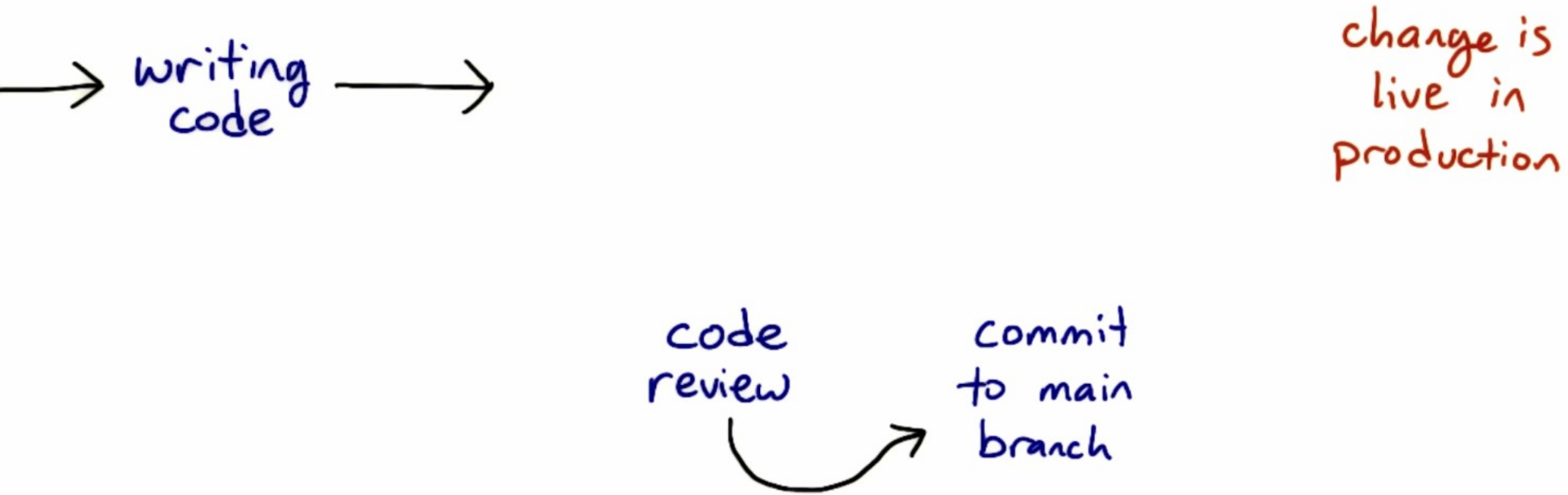


From Dev to Production

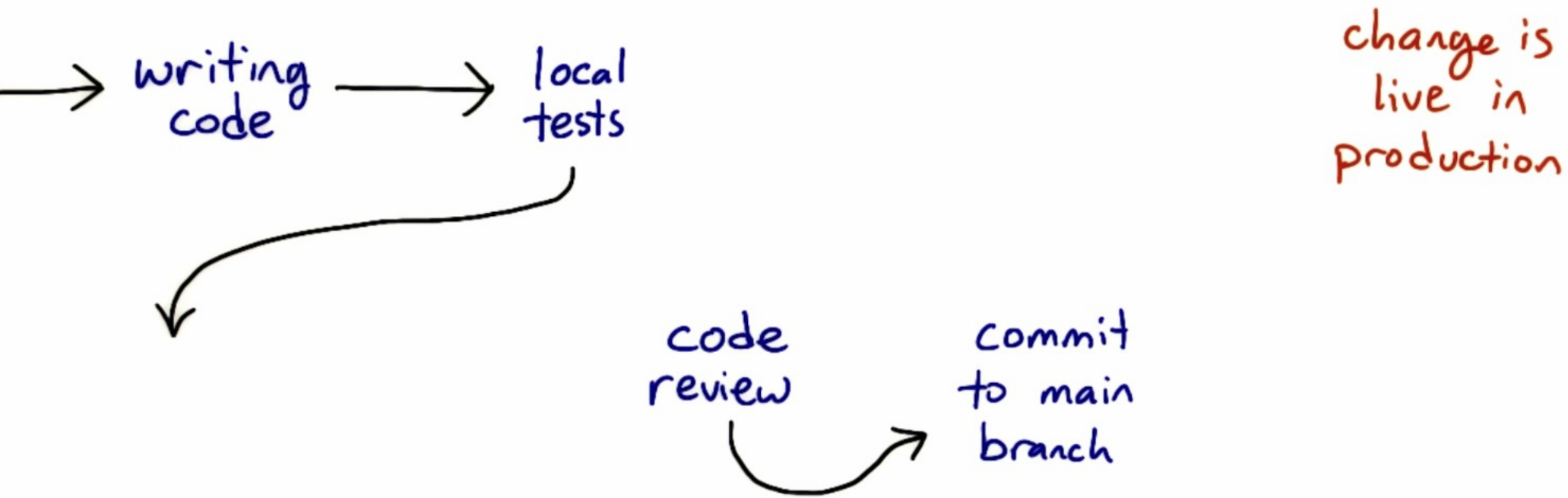




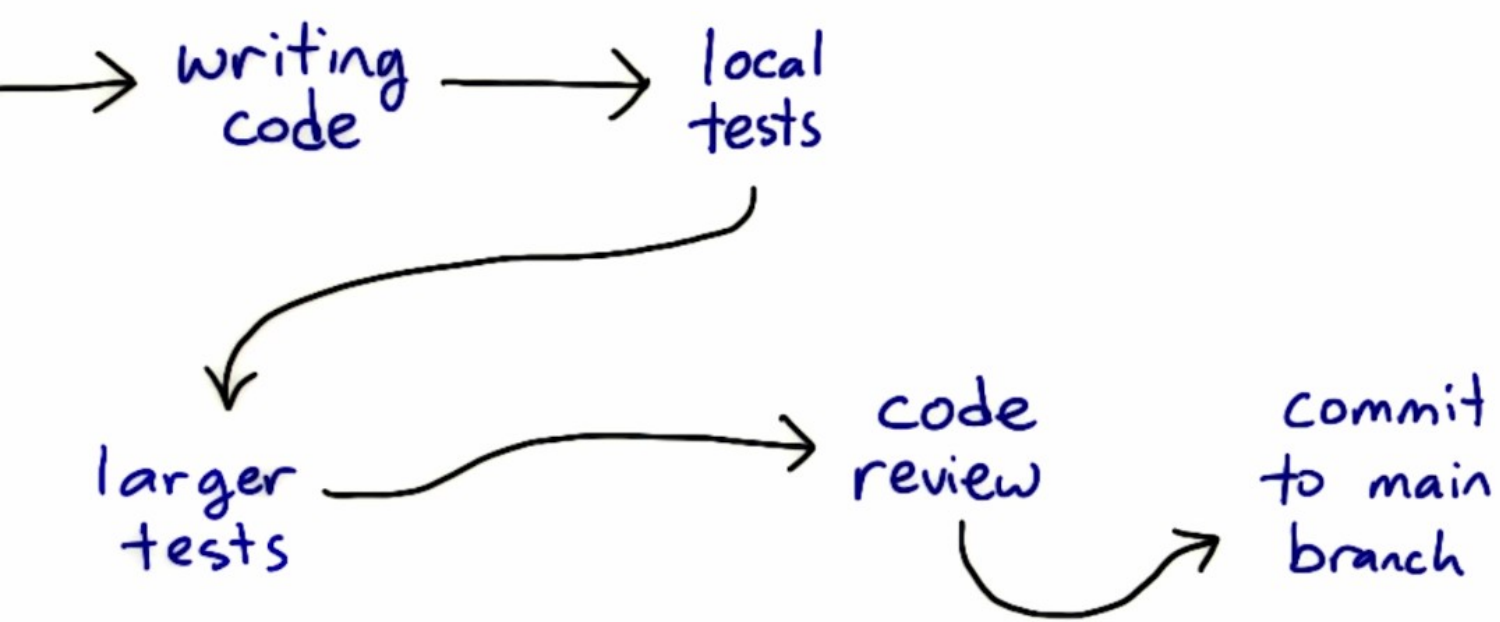
From Dev to Production



From Dev to Production

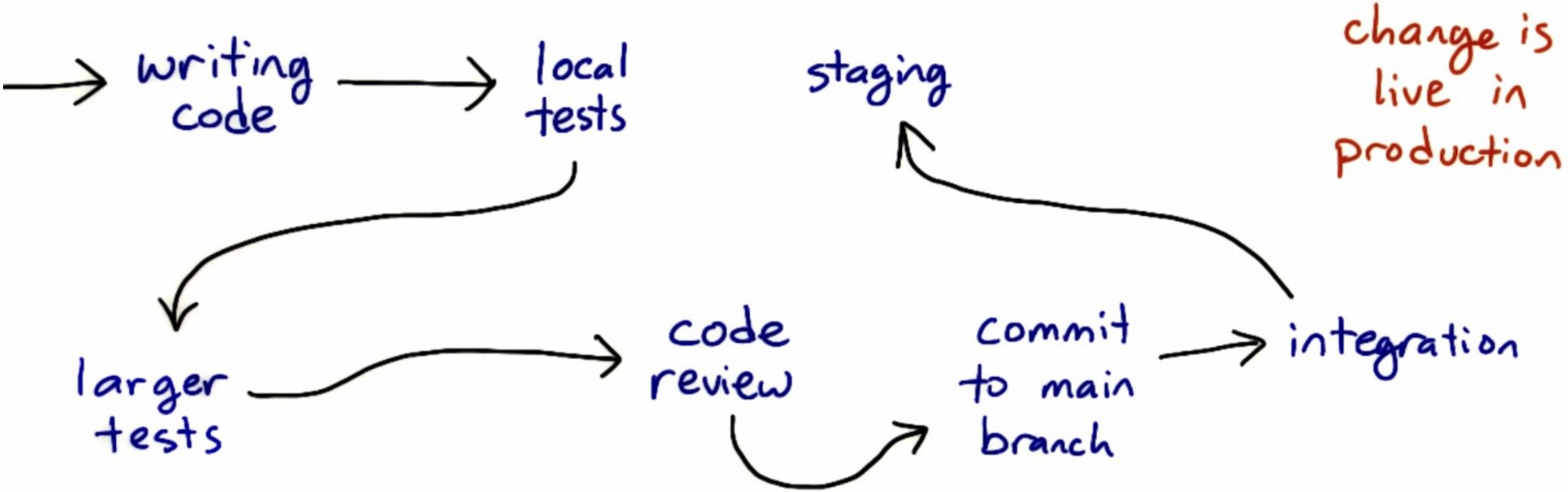


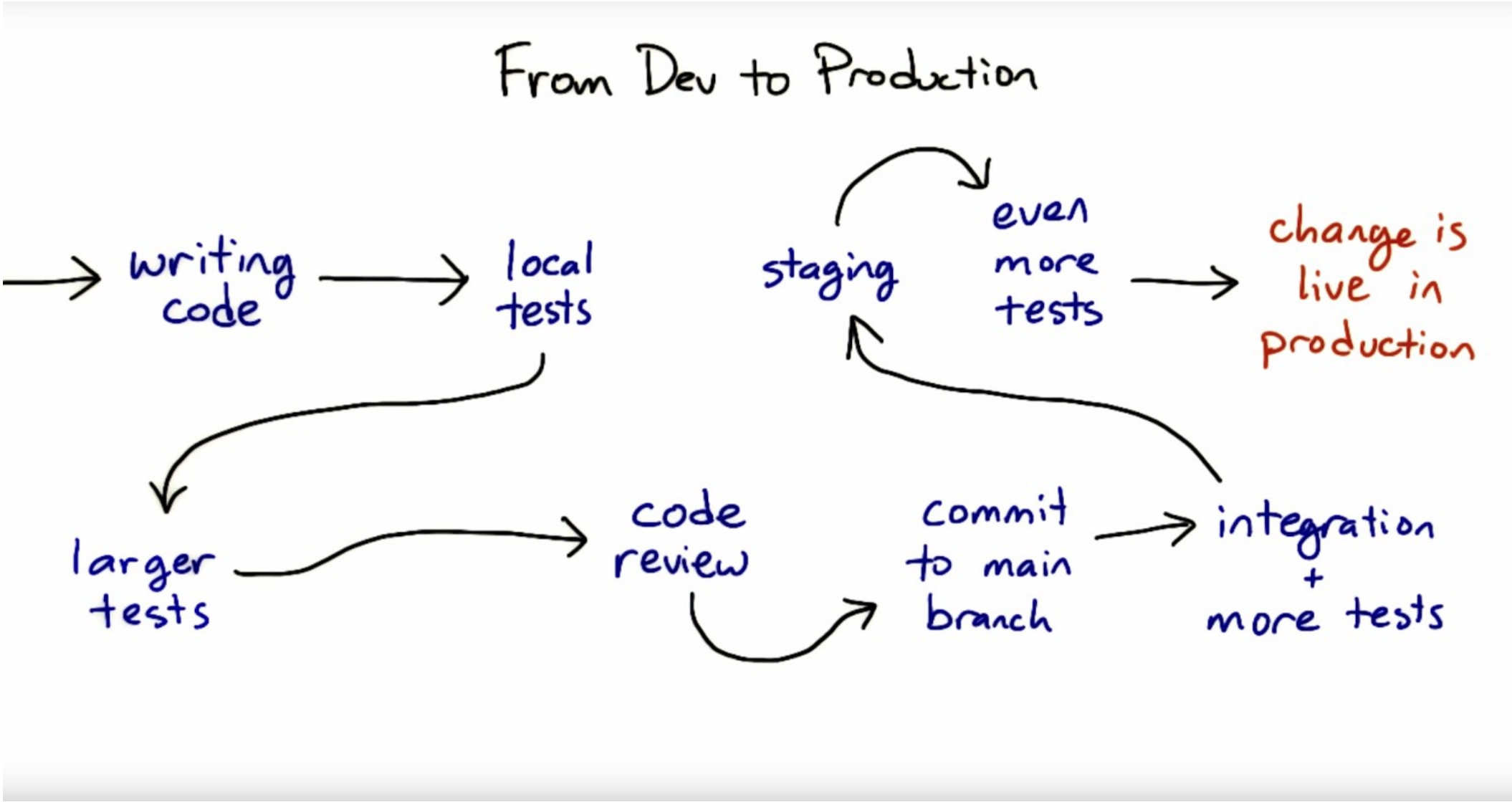
From Dev to Production



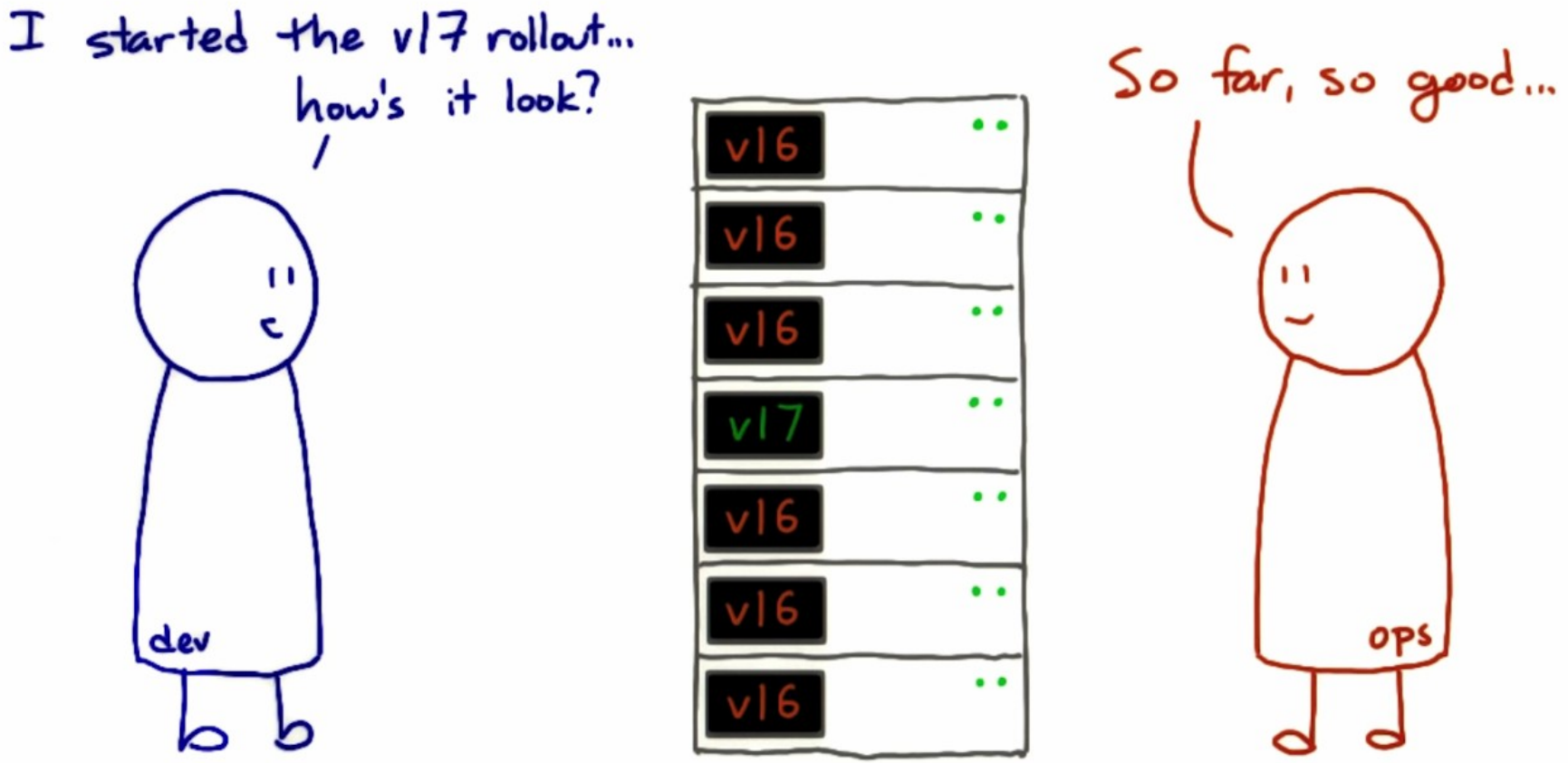
change is live in production

From Dev to Production

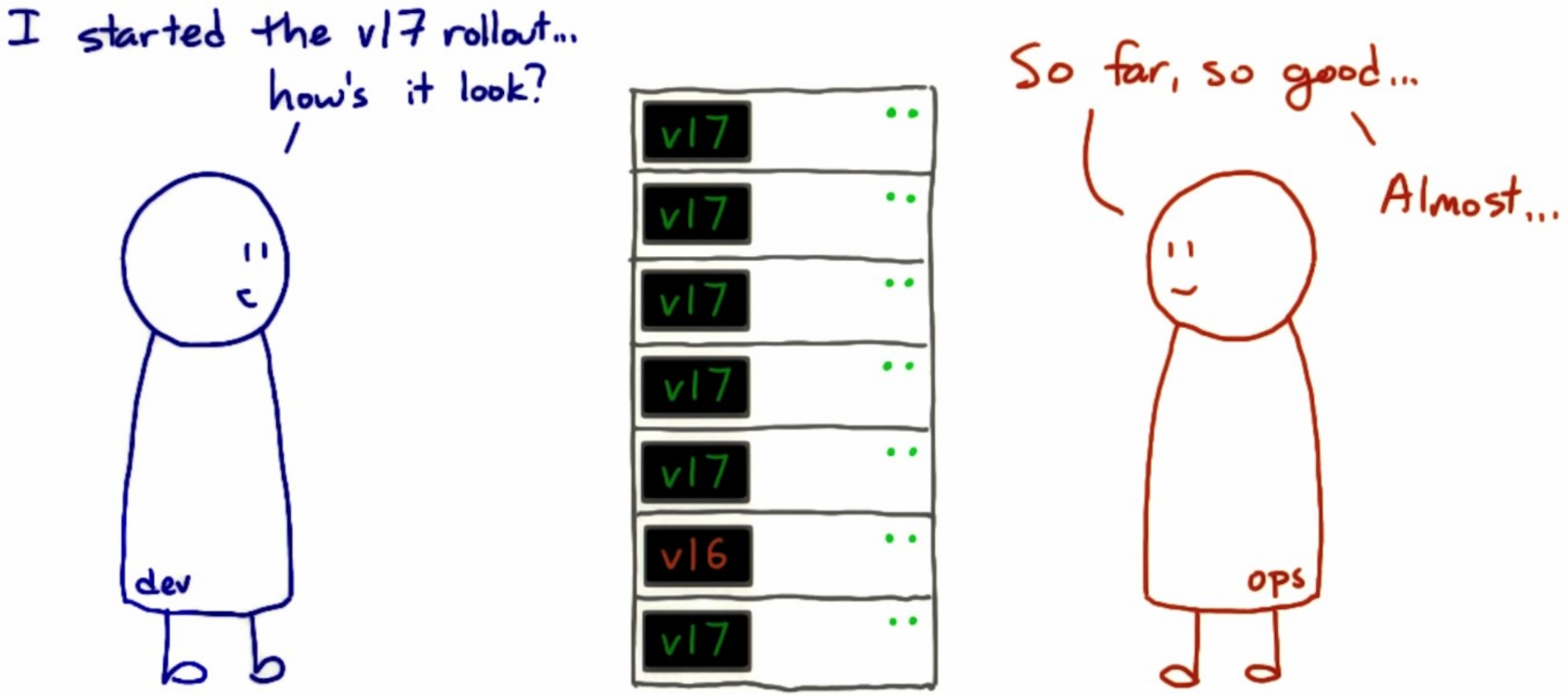



From Dev to Production
writing code
local tests
staging
even more tests
change is live in production
larger tests
code review
commit to main branch
integration + more tests

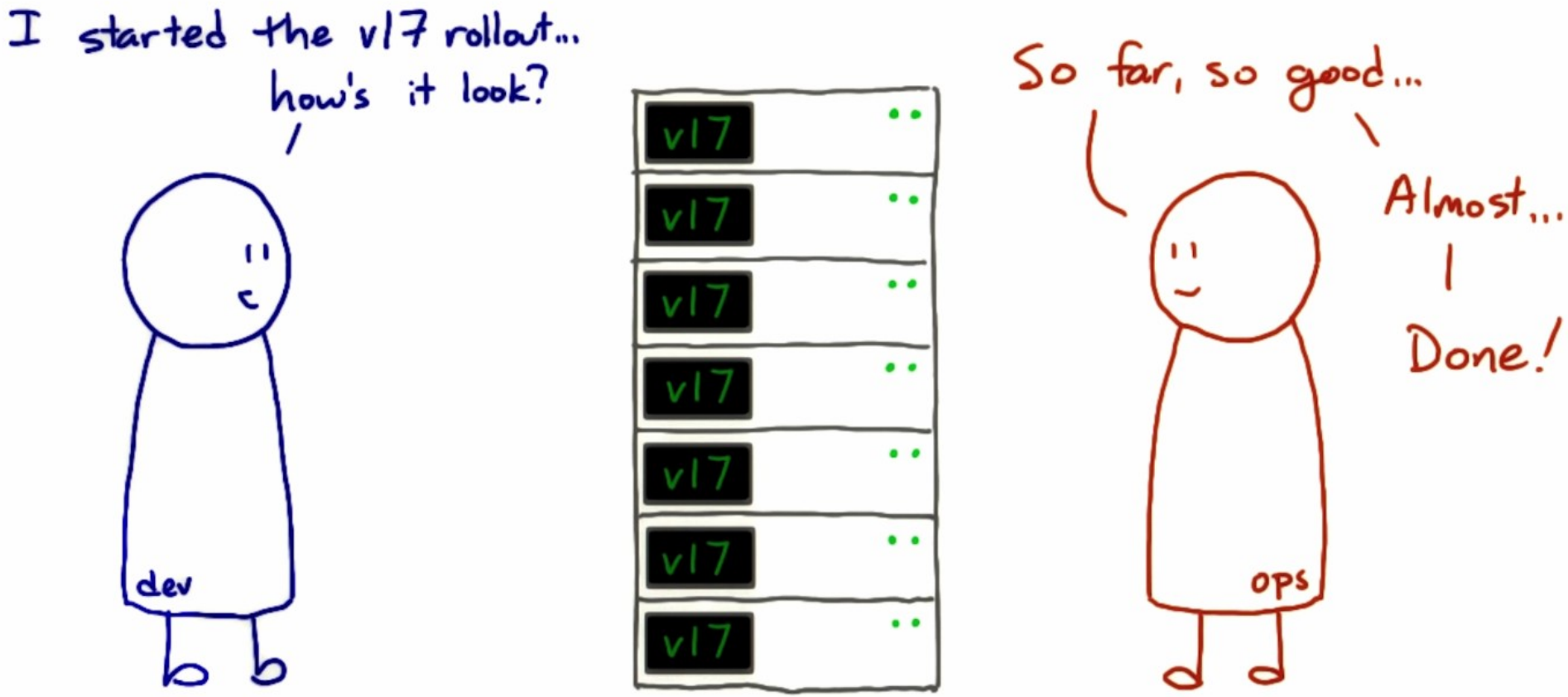
From Dev to Production



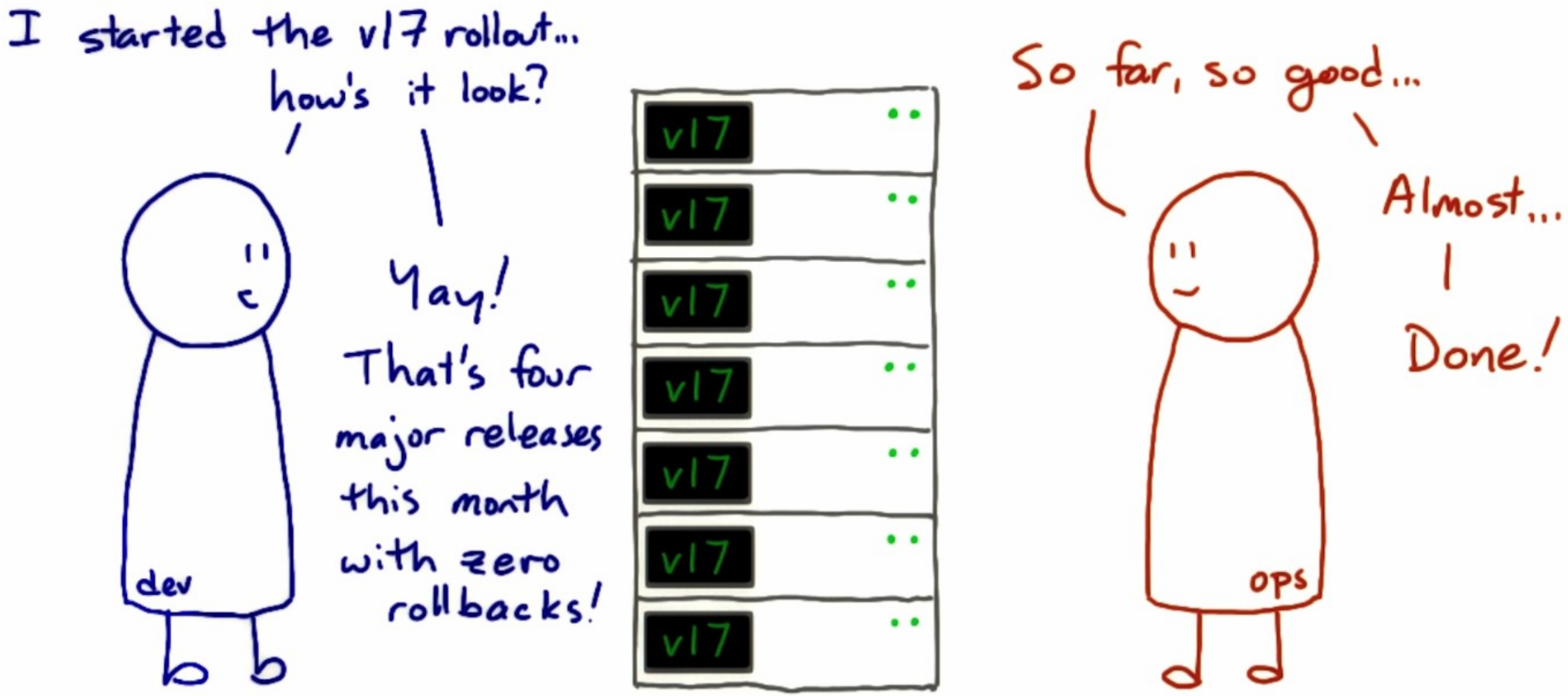
From Dev to Production

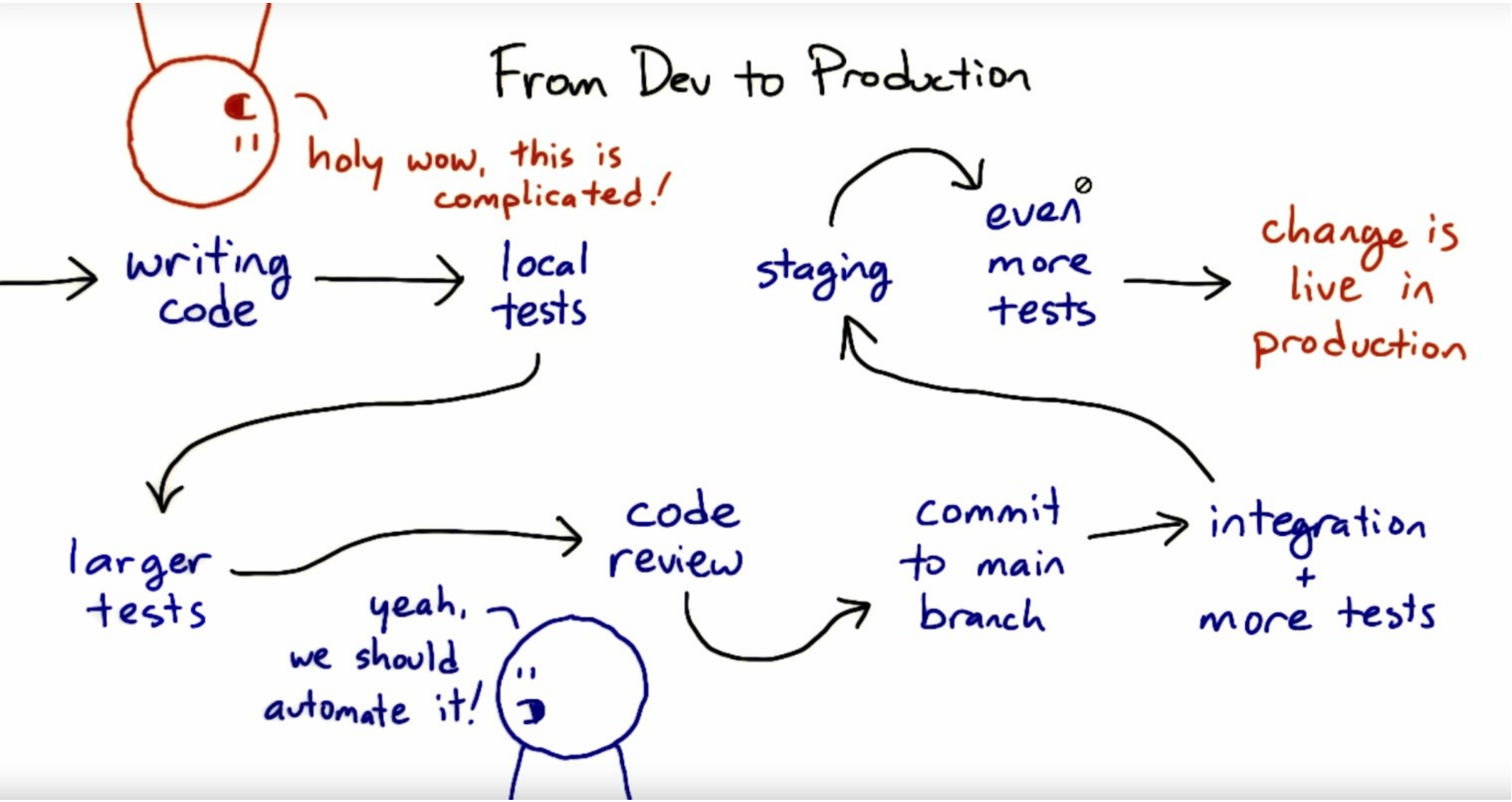


From Dev to Production

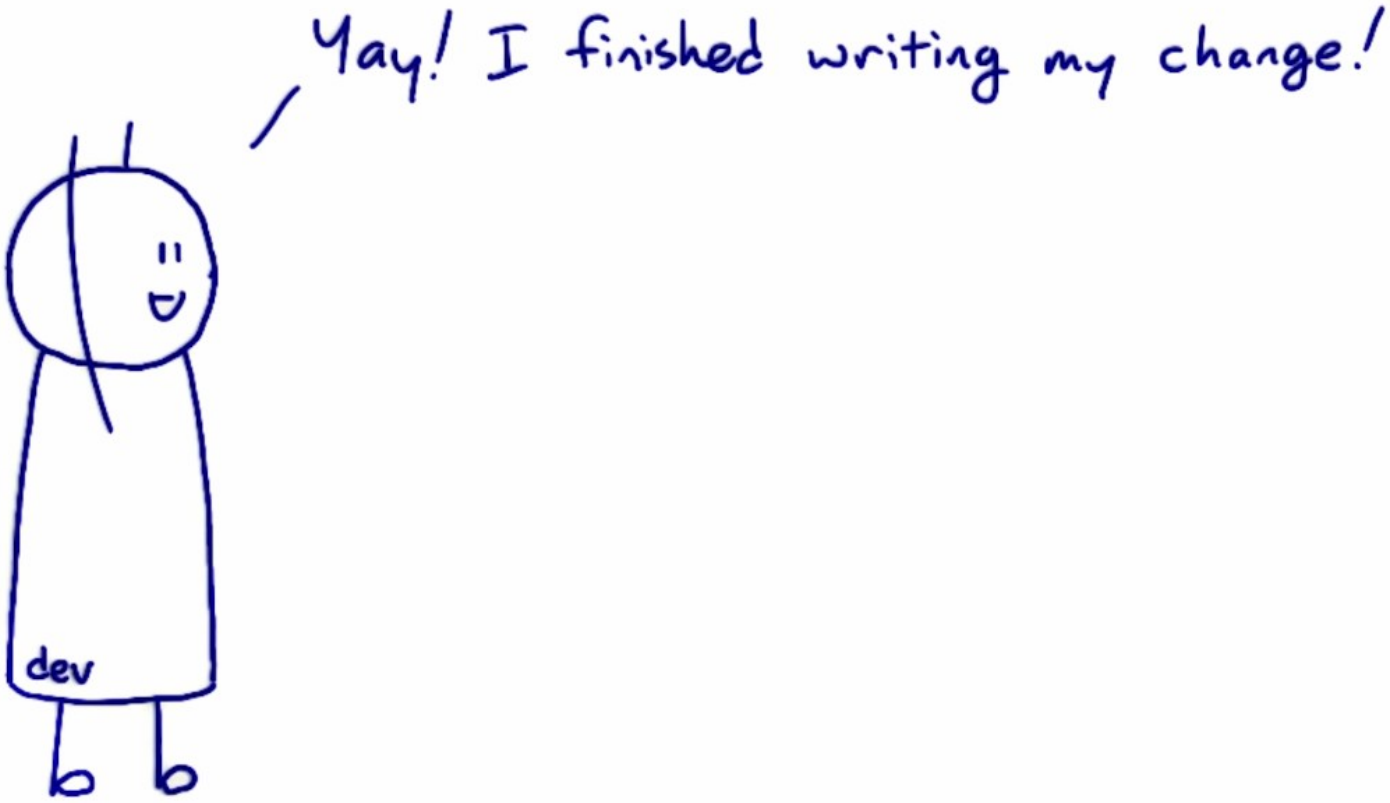


From Dev to Production

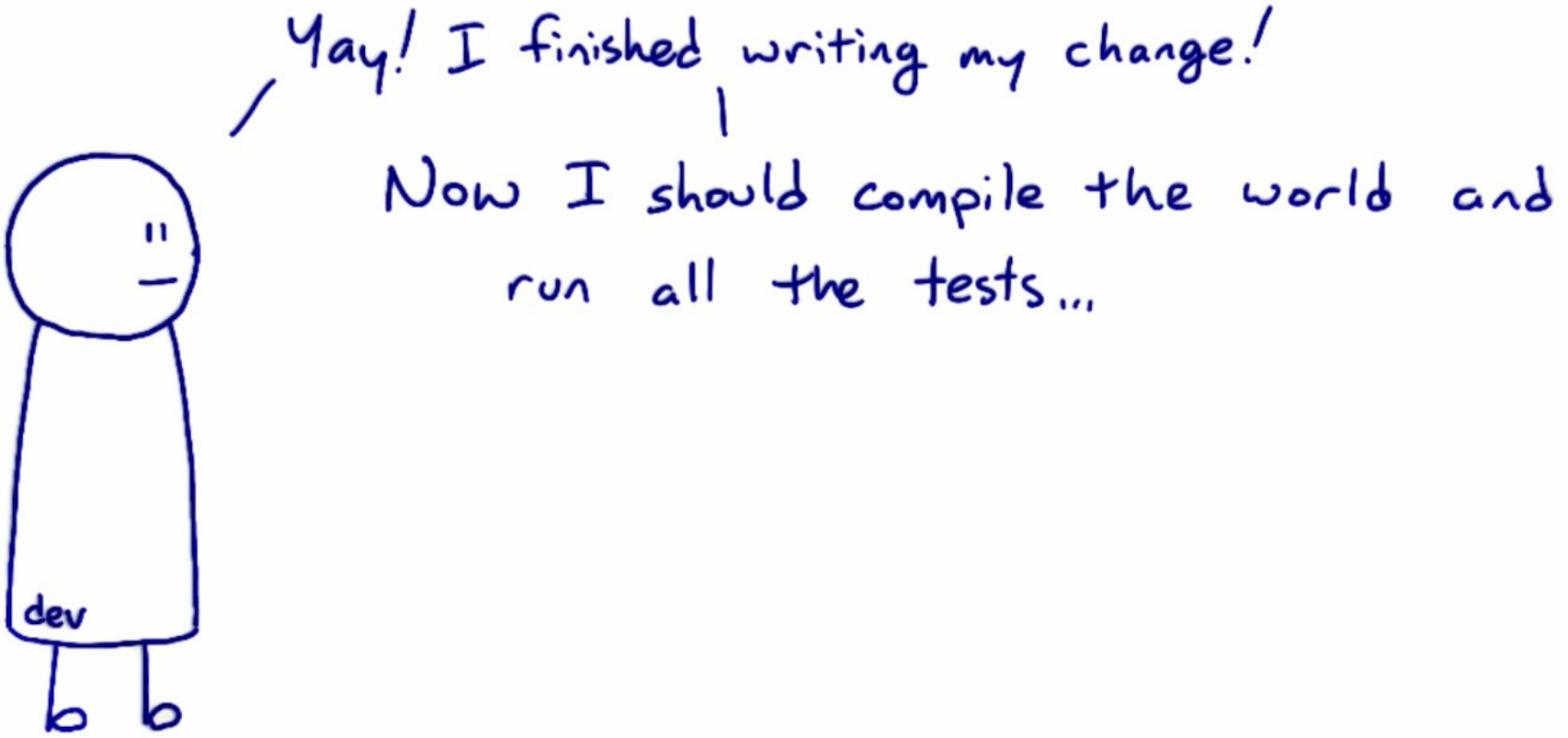



From Dev to Production
holy wow, this is complicated!
writing code
local tests
larger tests
code review
yeah, we should automate it!
commit to main branch
integration + more tests
staging
even more tests
change is live in production

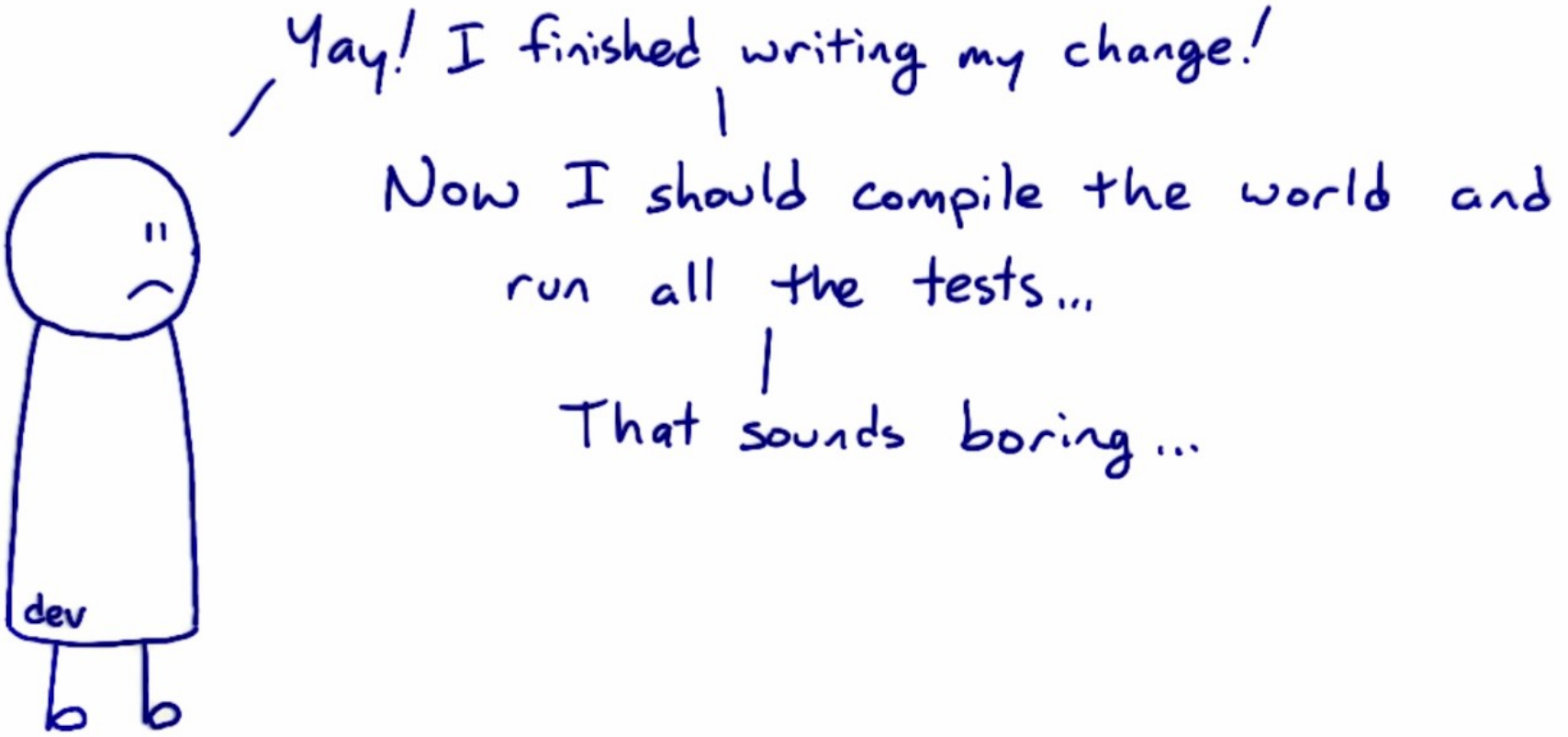
Continuous Integration



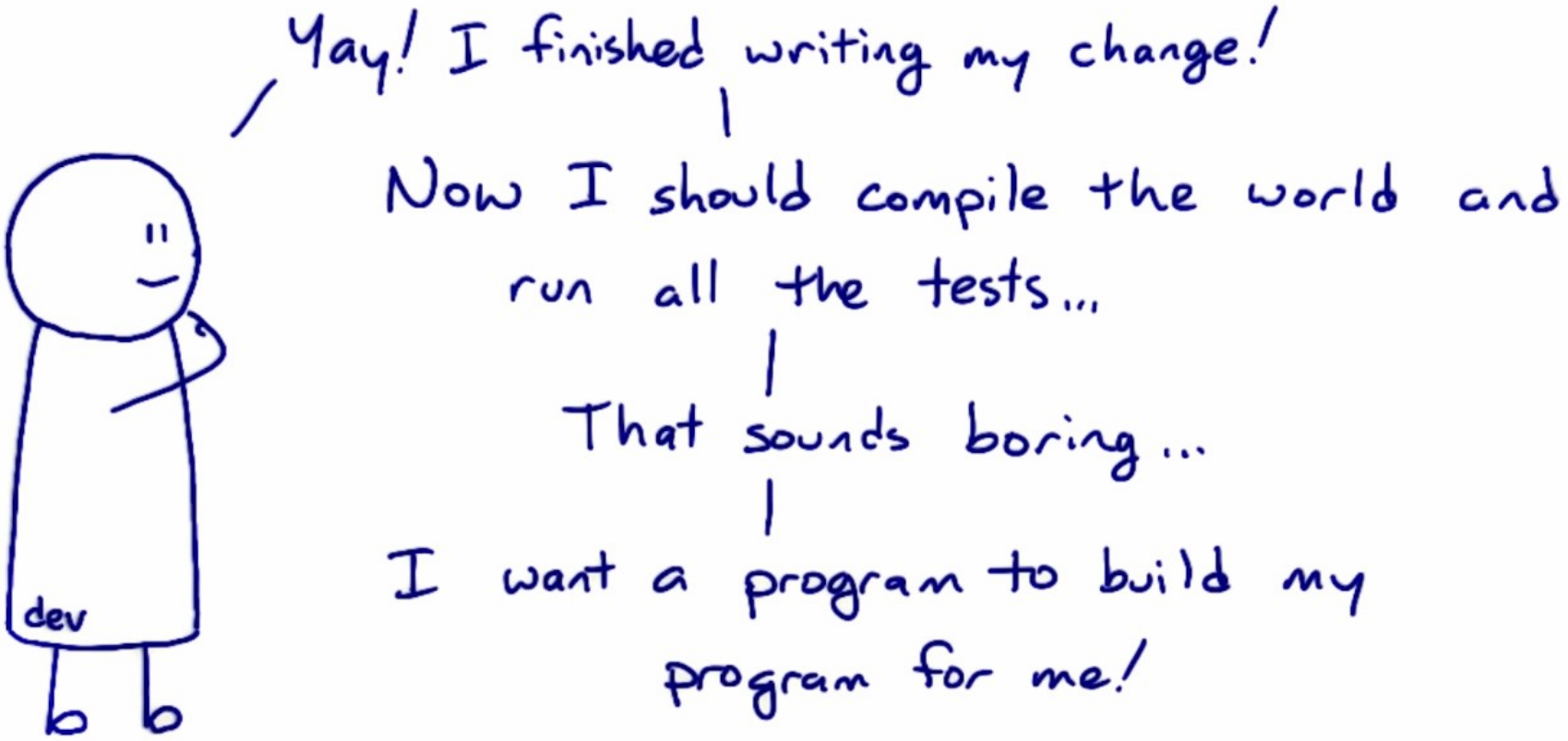
Continuous Integration



Continuous Integration



Continuous Integration



Continuous Integration

Jenkins

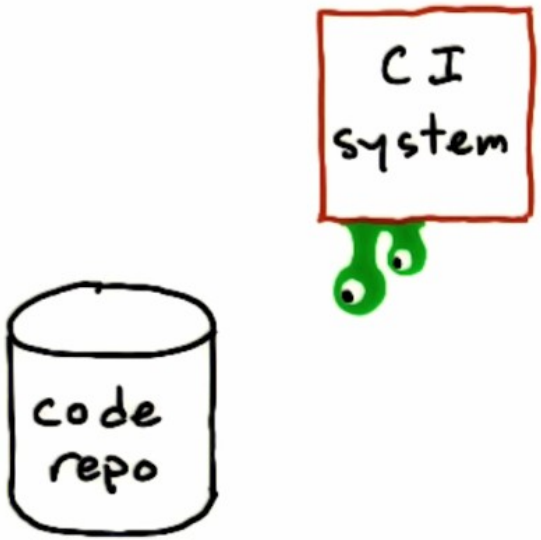
Continuous Integration

Jenkins

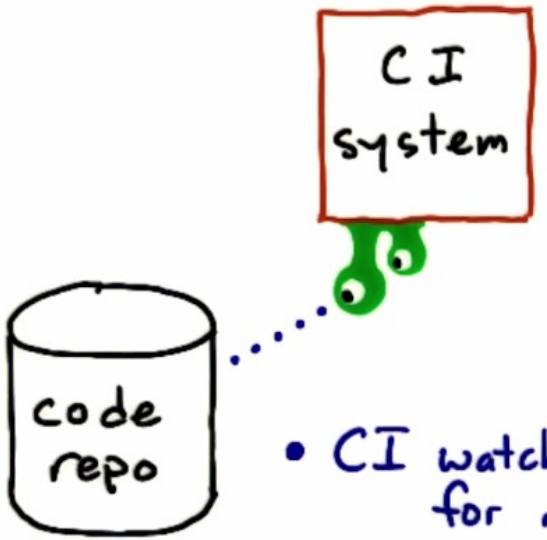
Travis CI

Circle CI

Continuous Integration

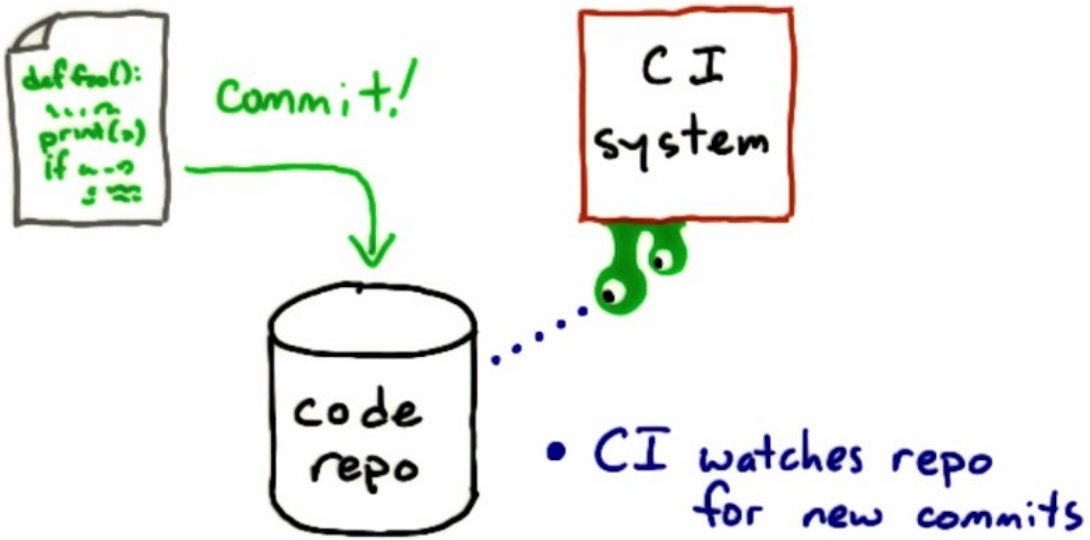


Continuous Integration



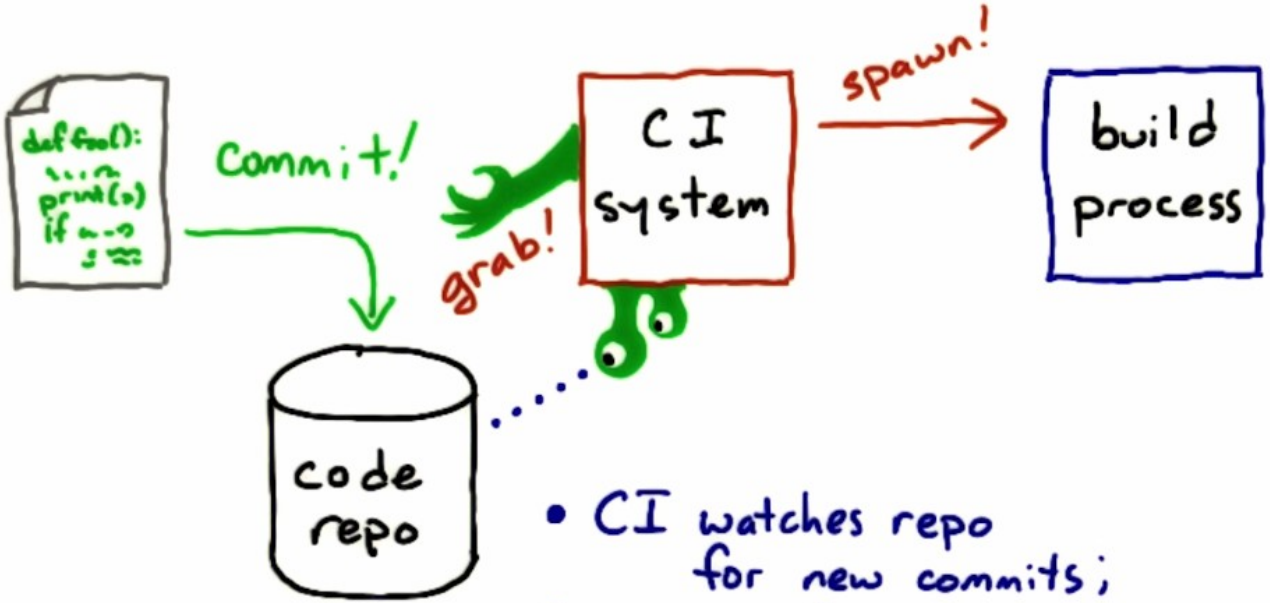
- CI watches repo for new commits

Continuous Integration



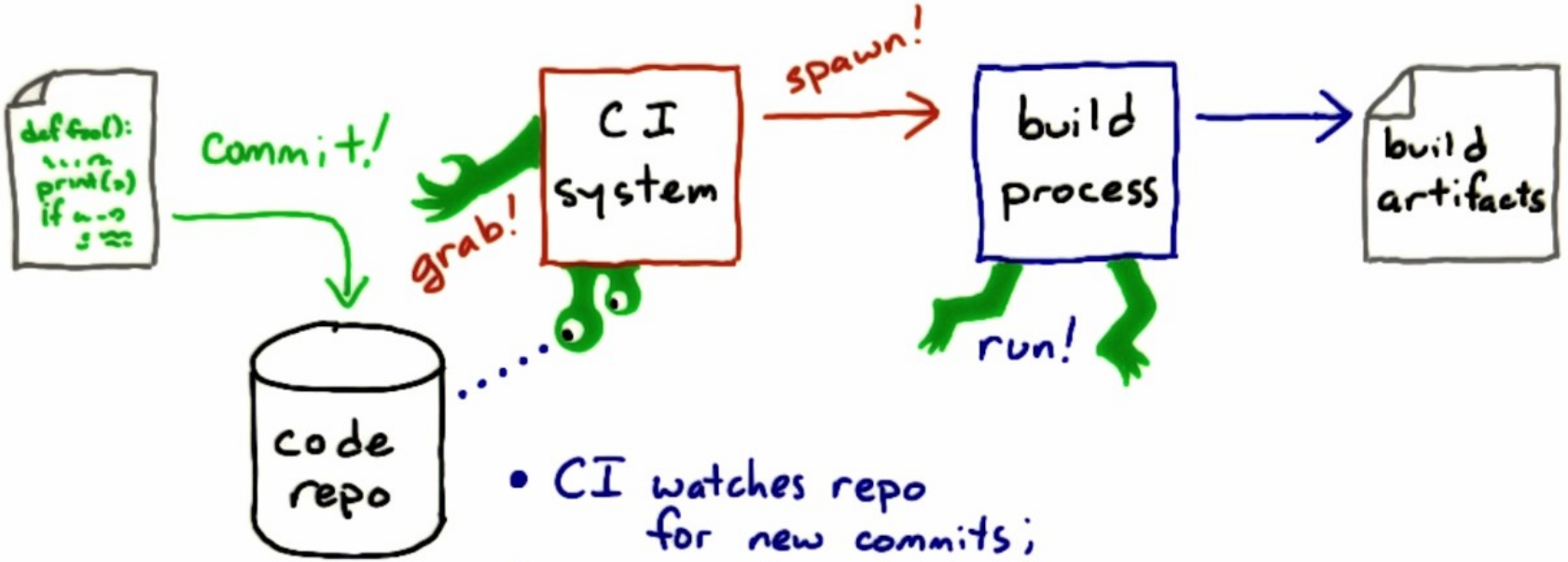
- CI watches repo for new commits

Continuous Integration



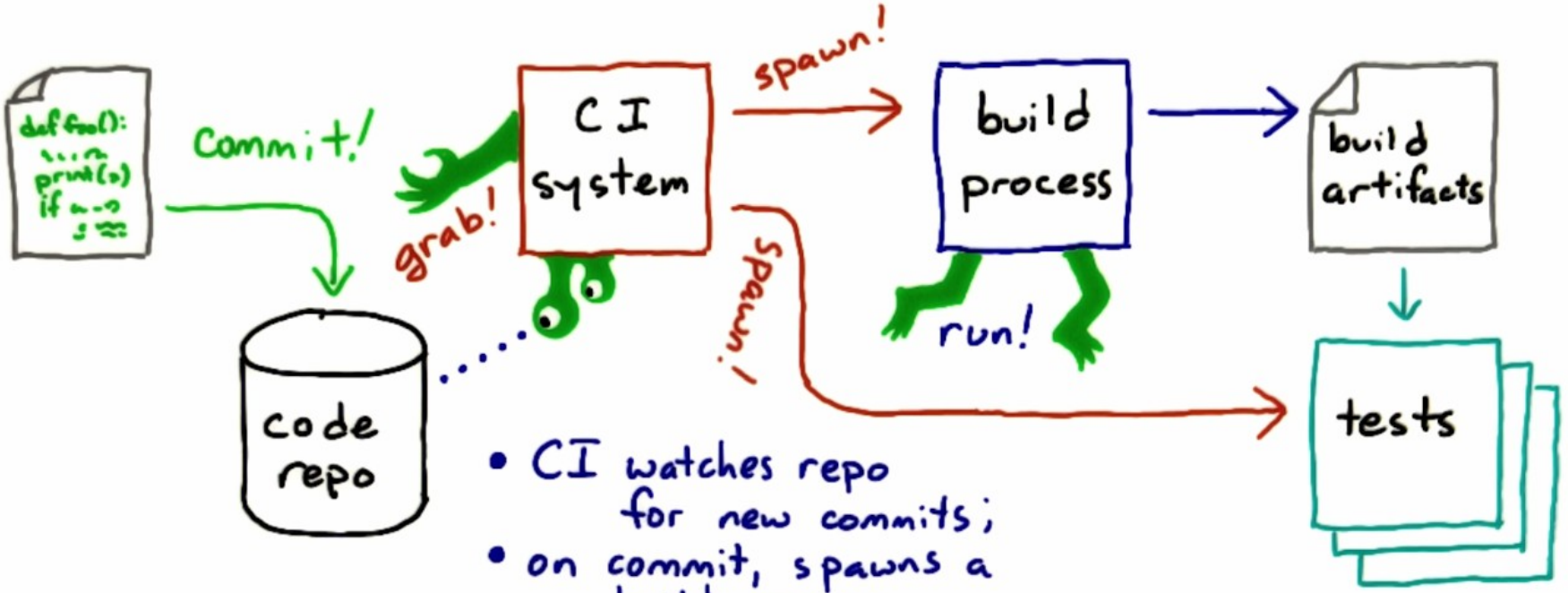
- CI watches repo for new commits;
- on commit, spawns a build process

Continuous Integration



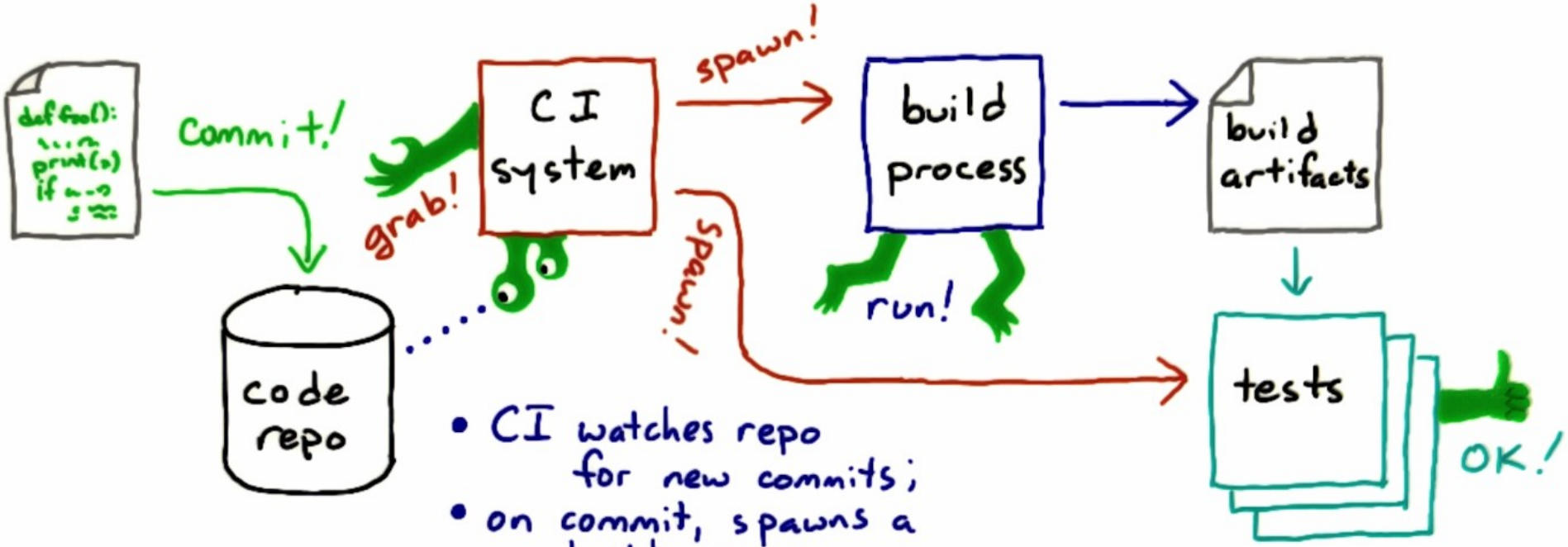
- CI watches repo for new commits;
- on commit, spawns a build process
- which runs and builds things

Continuous Integration



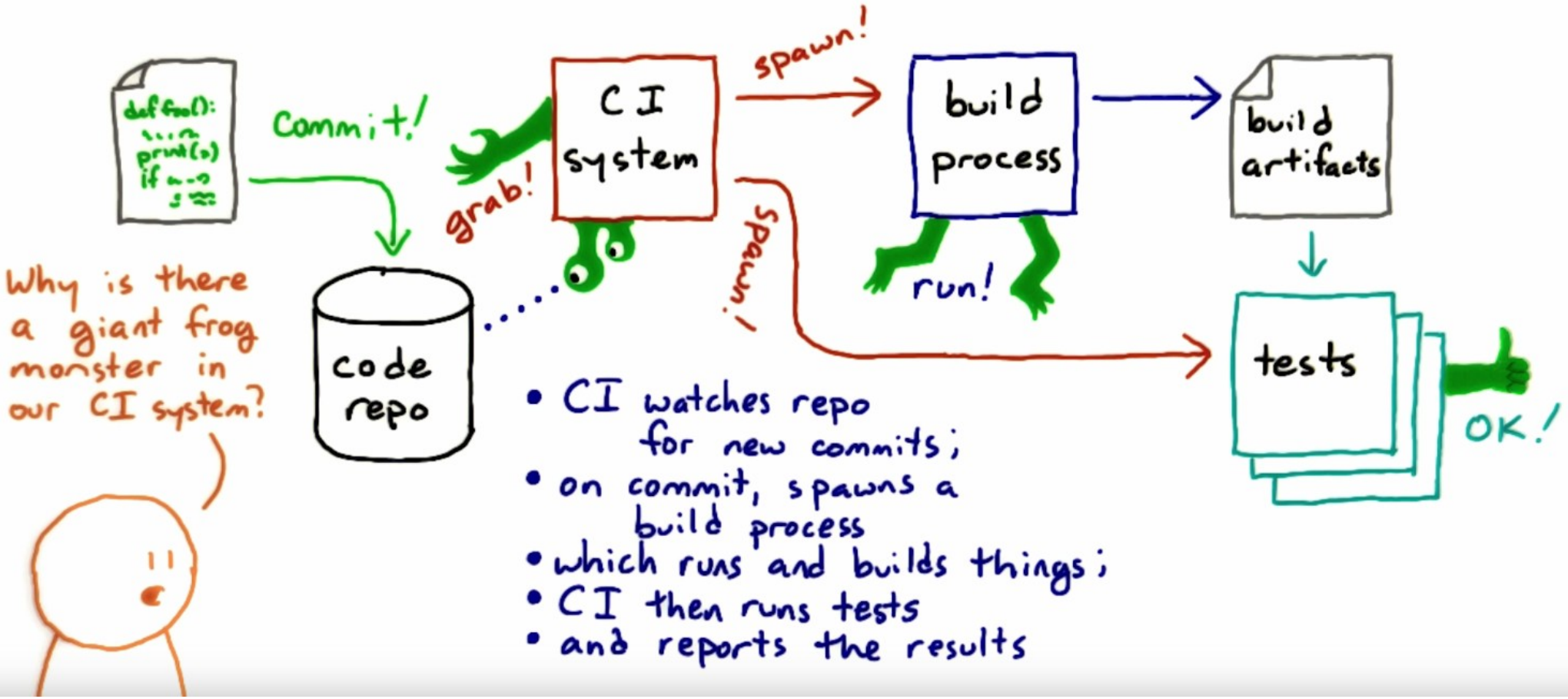
- CI watches repo for new commits;
- on commit, spawns a build process
- which runs and builds things;
- CI then runs tests

Continuous Integration



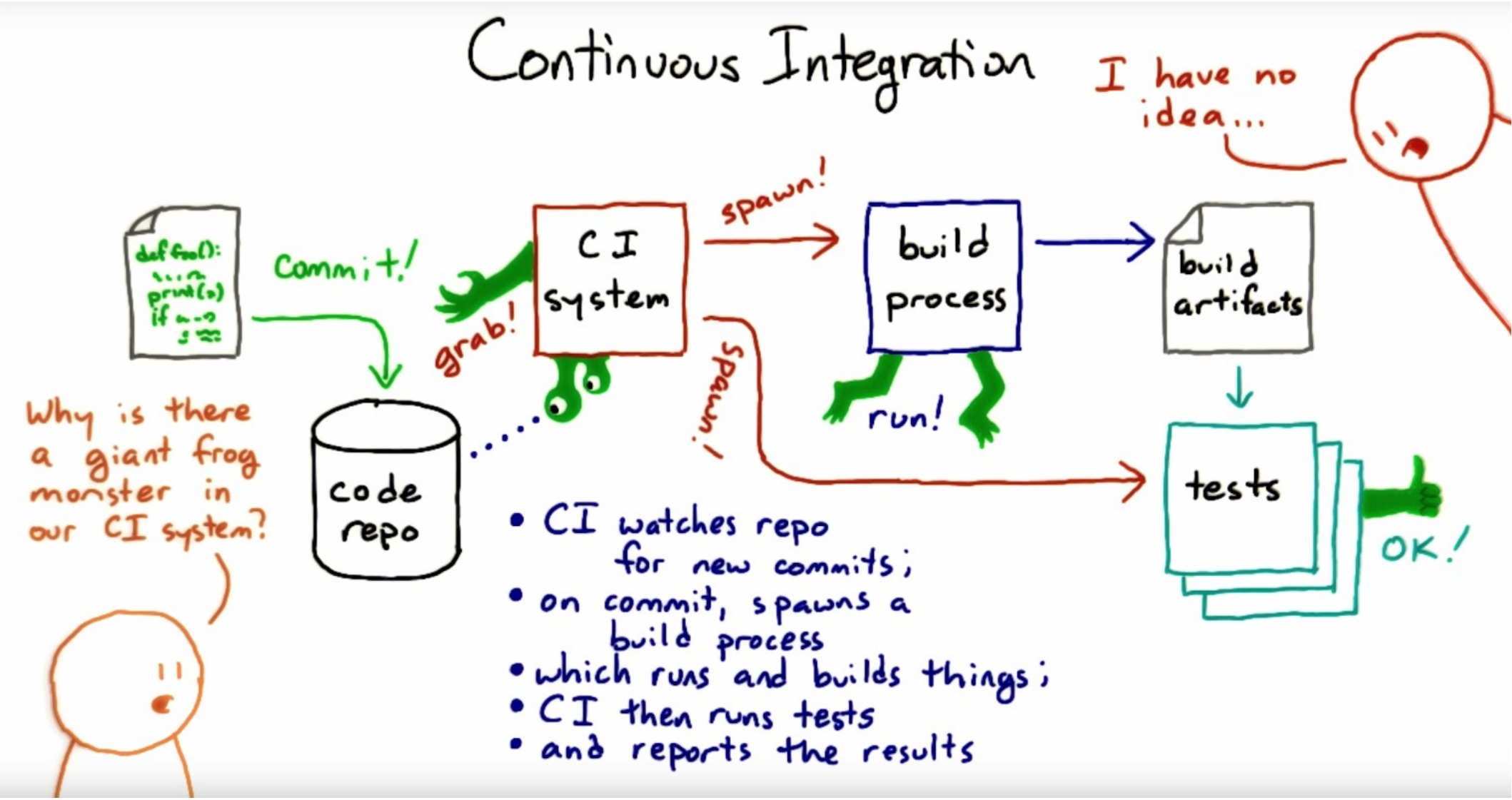
- CI watches repo for new commits;
- on commit, spawns a build process
- which runs and builds things;
- CI then runs tests
- and reports the results

Continuous Integration

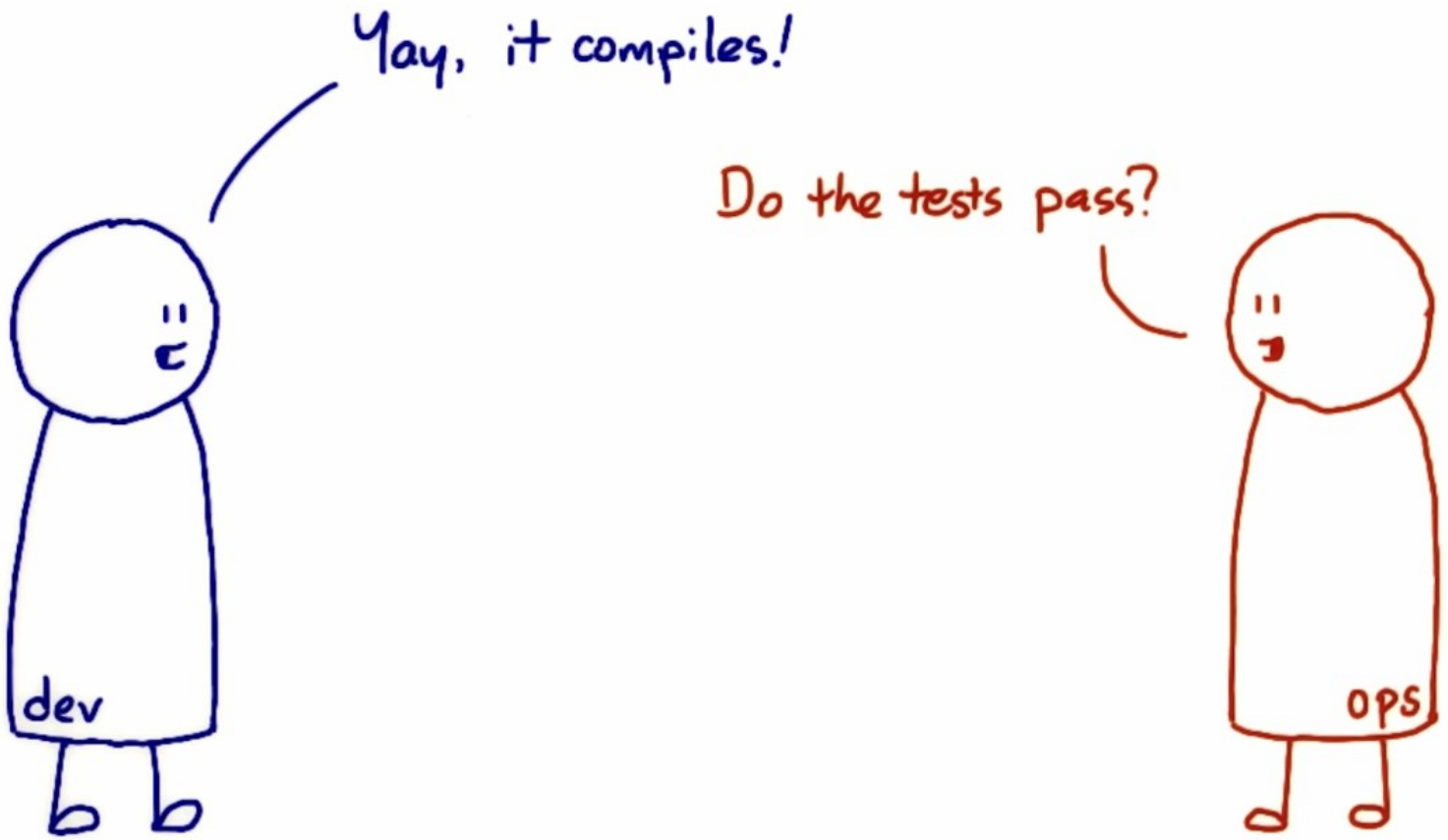


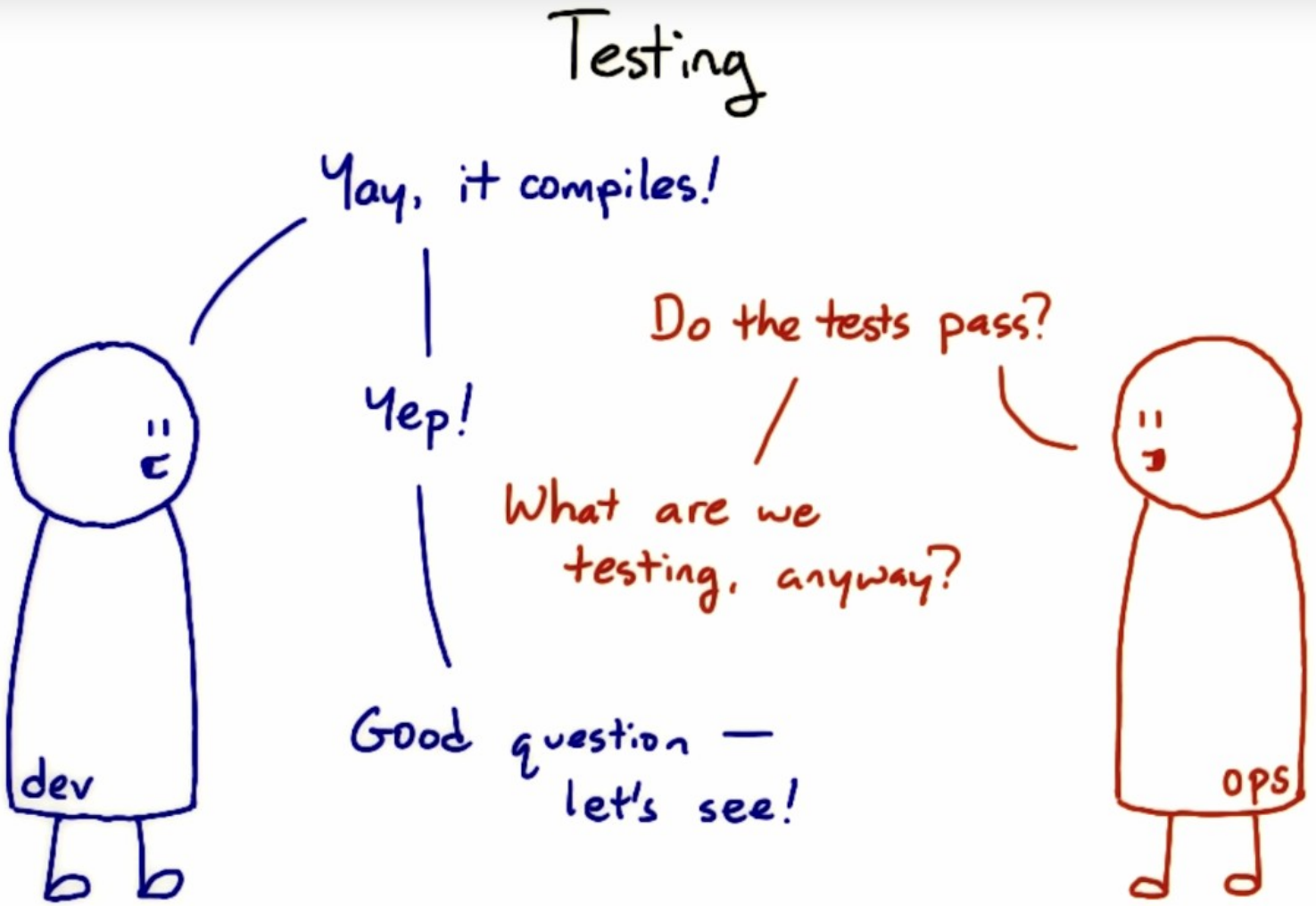
- CI watches repo for new commits;
- on commit, spawns a build process
- which runs and builds things;
- CI then runs tests
- and reports the results

Continuous Integration



Testing

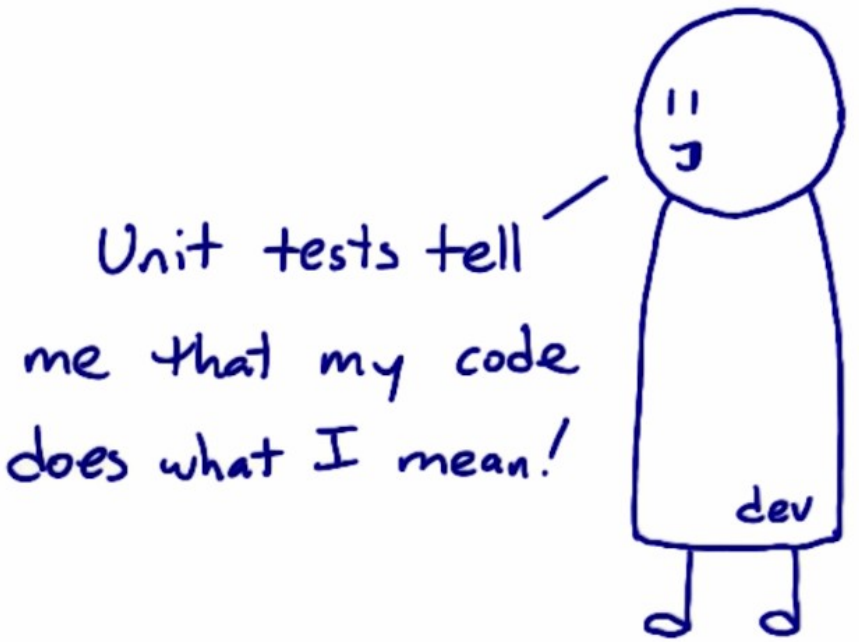



Testing
Yay, it compiles!
Yep!
Good question — let's see!
Do the tests pass?
What are we testing, anyway?
dev
ops

Testing

Unit testing

tests written alongside code, to test the behavior of individual units such as functions or classes.

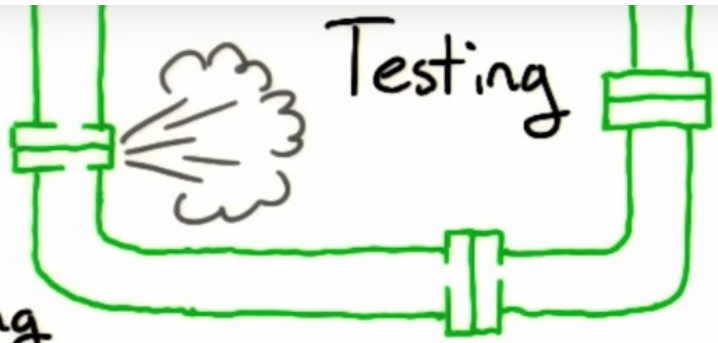


Testing

Unit testing

Regression testing

tests written as part of debugging, which verify that a bug is fixed. Kept in the test suite to ensure the bug is not reintroduced.



Unit testing

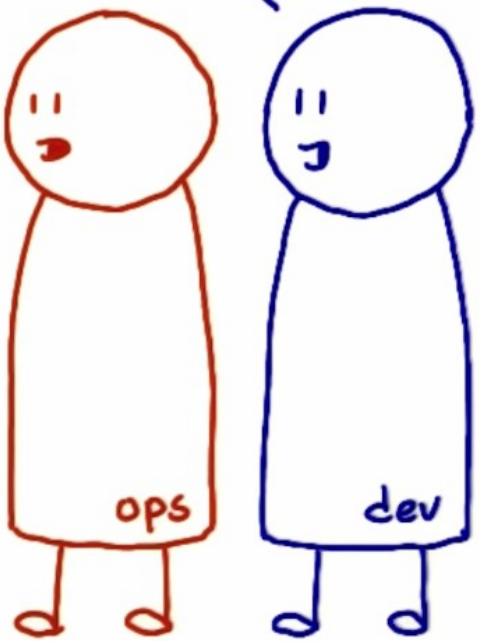
Regression testing

Smoke testing

Preliminary test of a system just after build, to make sure it runs at all — for instance doesn't crash on boot.

I call it a "build verification" test.

Smoke tests save us a lot of time by keeping broken builds out of the test pipeline.



Testing

Unit testing

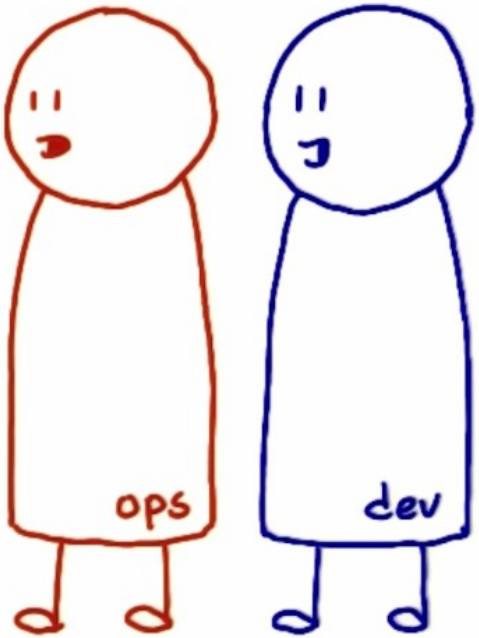
Regression testing

Smoke testing

System integration testing

Tests of a whole system, including dependencies such as databases or APIs, under a test load

Infrastructure-as-code makes it really easy to do system integration tests – the test can spin up an instance dynamically.



Testing

Unit testing

Regression testing

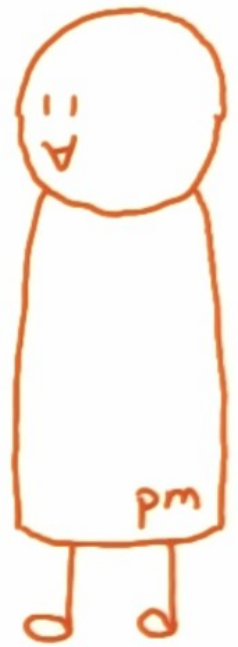
Smoke testing

System integration testing

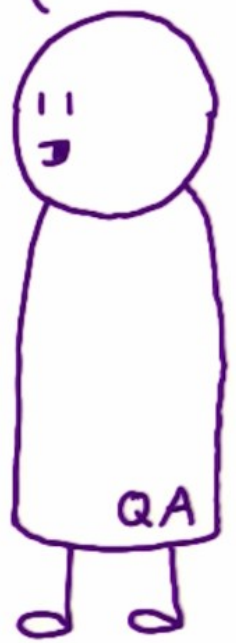
Automated acceptance testing

Scripted tests that verify that user-facing features work as planned.

We can even automate things like checking apps or web pages for accessibility requirements!



Testing

Unit testing
Regression testing
Smoke testing
System integration testing
Automated acceptance testing
Manual QA testing
    Approval process integrated with continuous deployment.

Before a launch goes out, we run it on the test instances and I check it over. If it looks good, I press the approval button.