

Continuous Delivery

and its design


YEGOR BUGAYENKO

Lecture #10 out of 16

90 minutes

All videos are in [this YouTube playlist](#).

All visual and text materials presented in this slidedeck are either originally made by the author or taken from public Internet sources, such as website. Copyright belongs to their respected authors.



Continuous Integration

Continuous Delivery

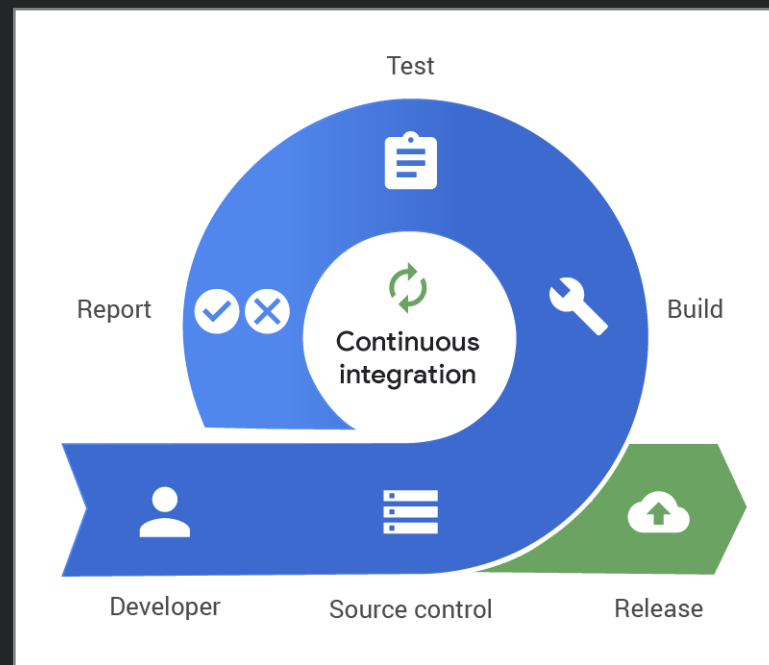
Open Source in GitHub

Books, Venues, Call-to-Action

Chapter #1:

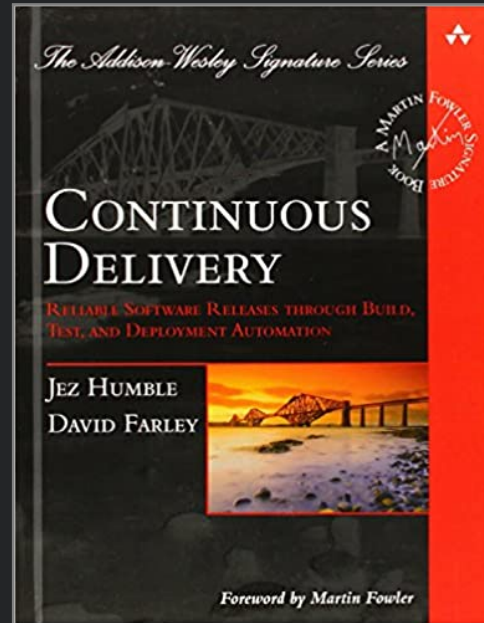
Continuous Integration

How Continuous Integration (CI) works:



It doesn't work :(

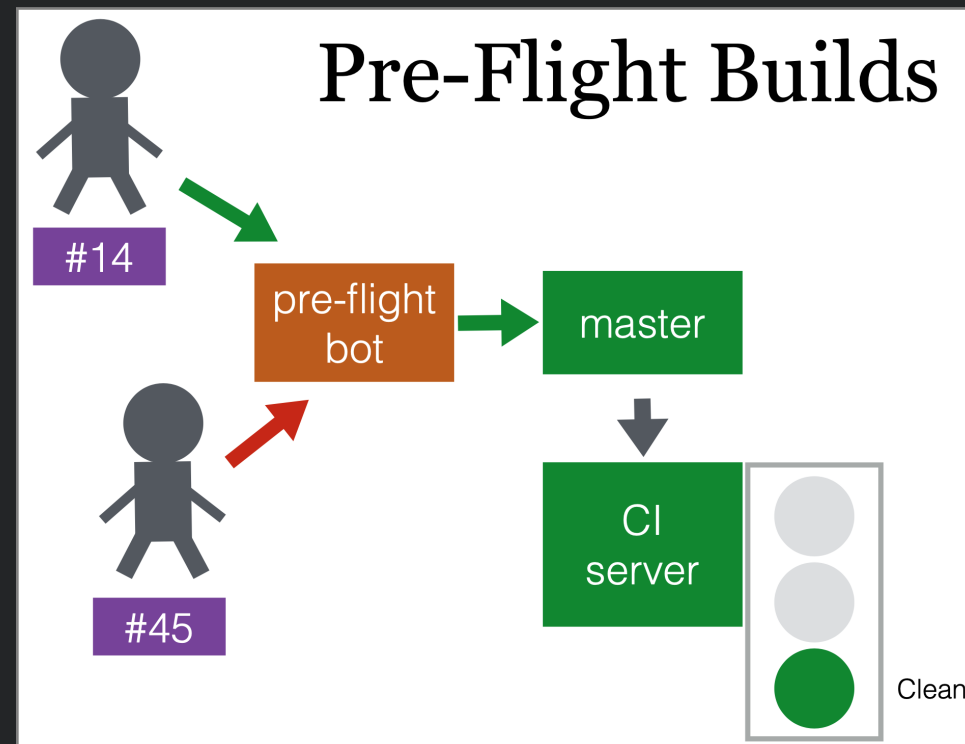
[[CI](#) Pre-Flight]



“Crucially, if the build fails, the development team stops whatever they are doing and fixes the problem immediately”

— Jez Humble, Continuous Delivery, p. 55

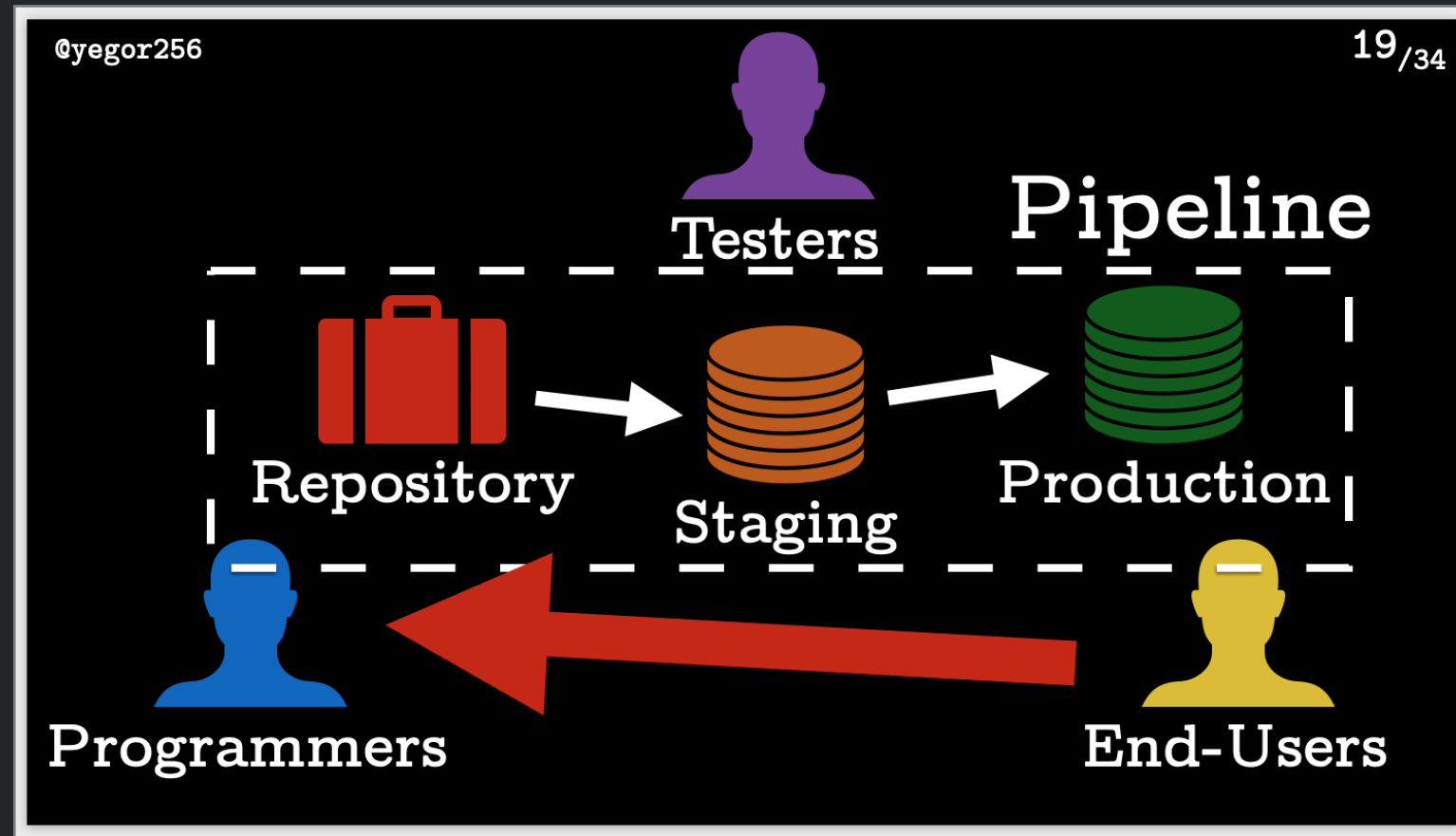
Pre-Flight Builds



Master branch is “read-only”!

Chapter #2:

Continuous Delivery



youtube.com/
nwAbuDrV8 →

Joker Conf 2018, Quality vs Quantity, in Russian



“Each stage of a build pipeline is looking for reasons to reject the build. Tests failed? Reject it. Lint complains? Reject it. Build fails integration tests in staging? Reject it. Finished archive smells funny? Reject it.”

— Michael Nygard, Release It!

Quality Wall

Linters

Static analyzers

Unit tests

Integration tests

Test coverage control

Mutation coverage control

Manual code review(s)

Target Platforms

Static web site: GitHub Pages

Library: Maven Central, RubyGems.org, Npm.org, etc.

Small web app: Heroku or Dokku

Bigger web app: AWS Elastic Beanstalk

Mobile app: TestFlight

Versioning

snapshot, alpha, beta, final

semver.org: 1.4.17

0ver.org: 0.43.3

CI/CD Tools

Jenkins

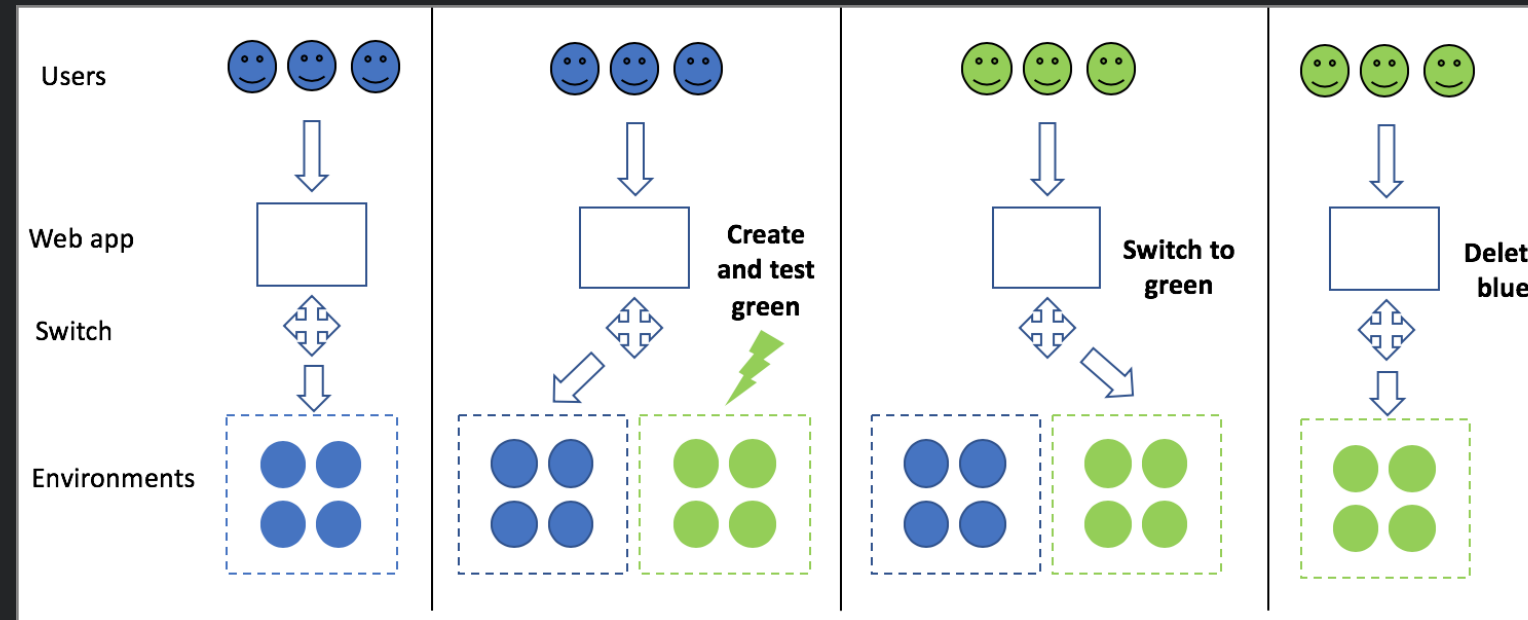
1000+ Hosted Services like Travis

GitHub Actions

Rultor.com

[Wall Targets Versions Tools [Blue/Green](#)]

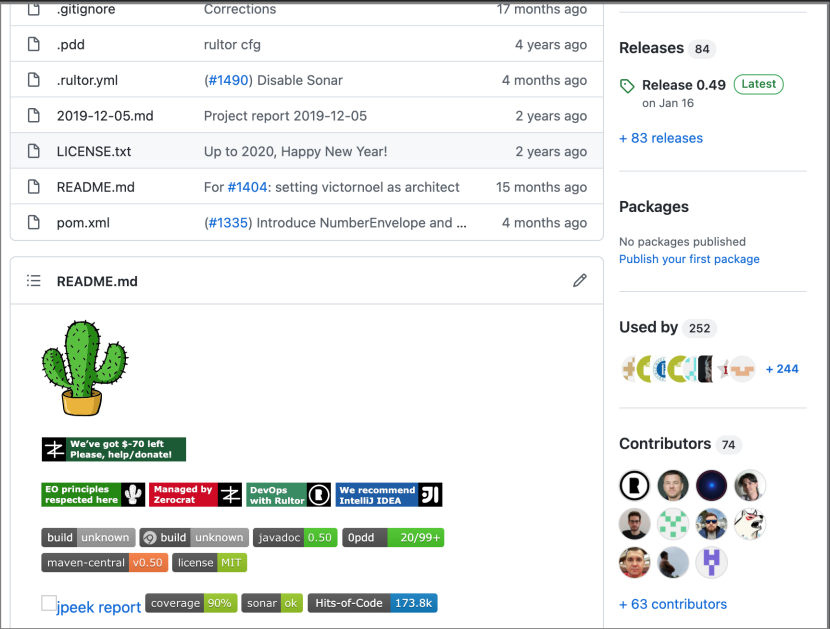
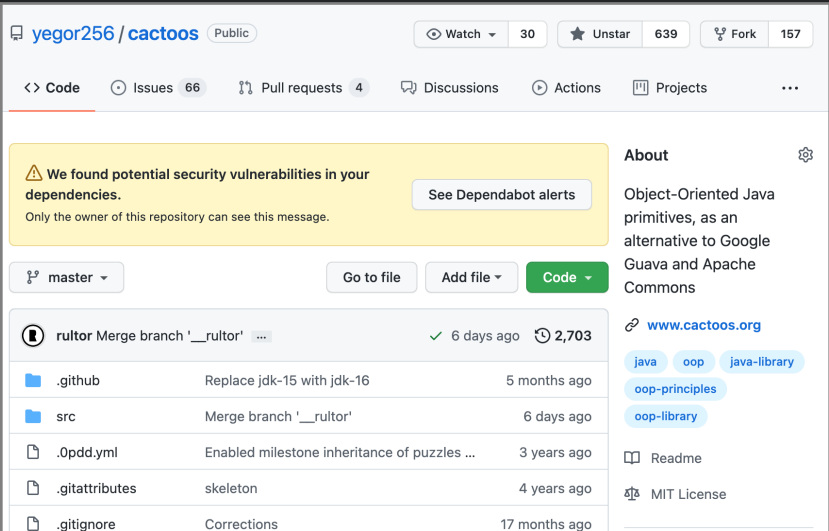
Blue/Green Deployment



May not work with databases :(

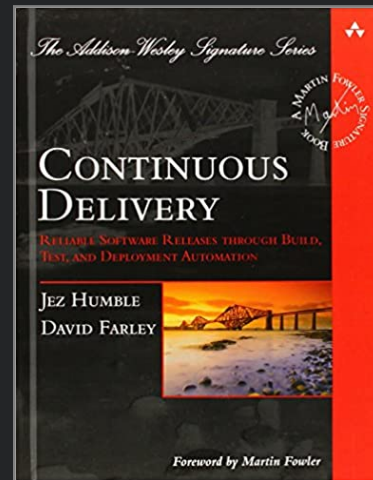
Chapter #3:

Open Source in GitHub



Chapter #4:

Books, Venues, Call-to-Action



“Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation” by JEZ HUMBLE ET AL.



“Release It!: Design and Deploy Production-Ready Software” by MICHAEL T. NYGARD

Where to go:

Collect 1000 stars on GitHub for your product.

Call to Action:

Setup continuous delivery pipeline in your application so that it is released to production automatically on each commit to `master` branch.

Still unresolved issues:

- How to parallelize tests?
- How to speed up merging into “master”?
- How to generate tests automatically?
- How to enforce quality for legacy code base?