

Continuous integration and automated testing

Riivo Kikas

University of Tartu

riivokik@ut.ee

Today

- Overview of Continuous Integration
- Automated testing
- Demo of CI in action – Web Apps and CircleCI
 - Spring and Java
 - PHP

Modern software development



IDEA



ISSUE



PLAN



CODE



COMMIT



TEST



REVIEW



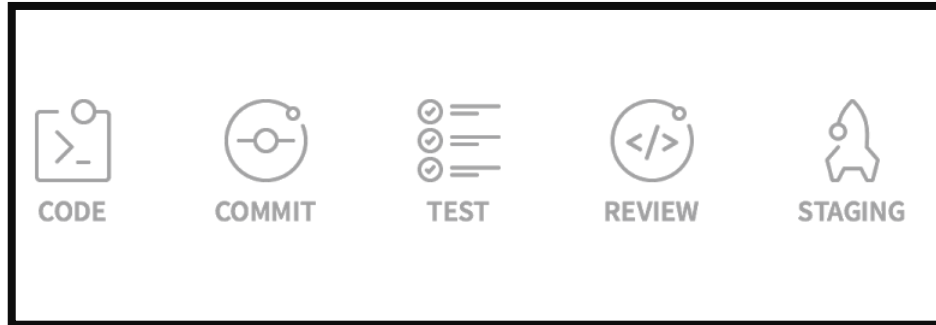
STAGING



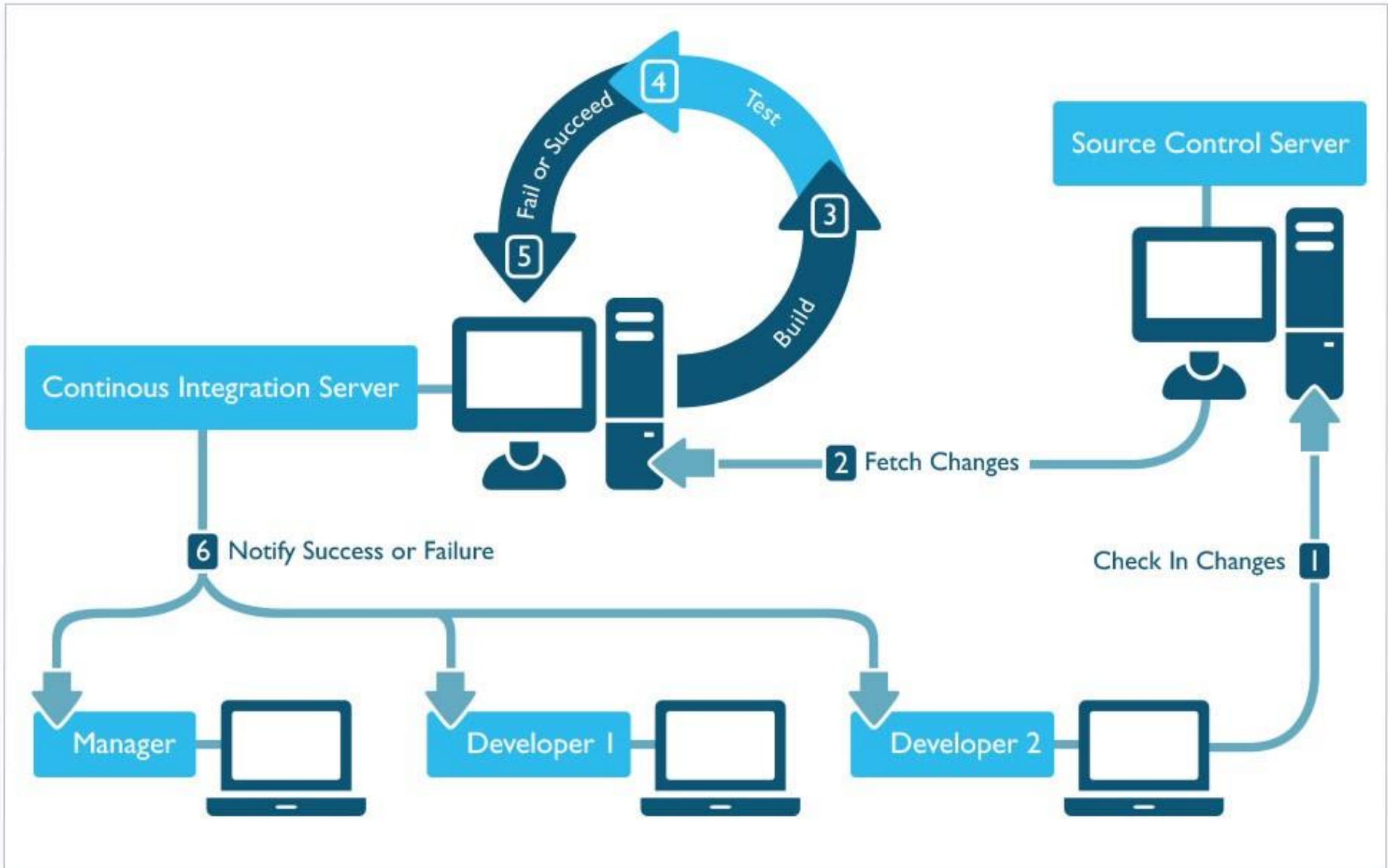
PRODUCTION



FEEDBACK



Aim for frequent releases of quality software, reduce repetitive tasks!

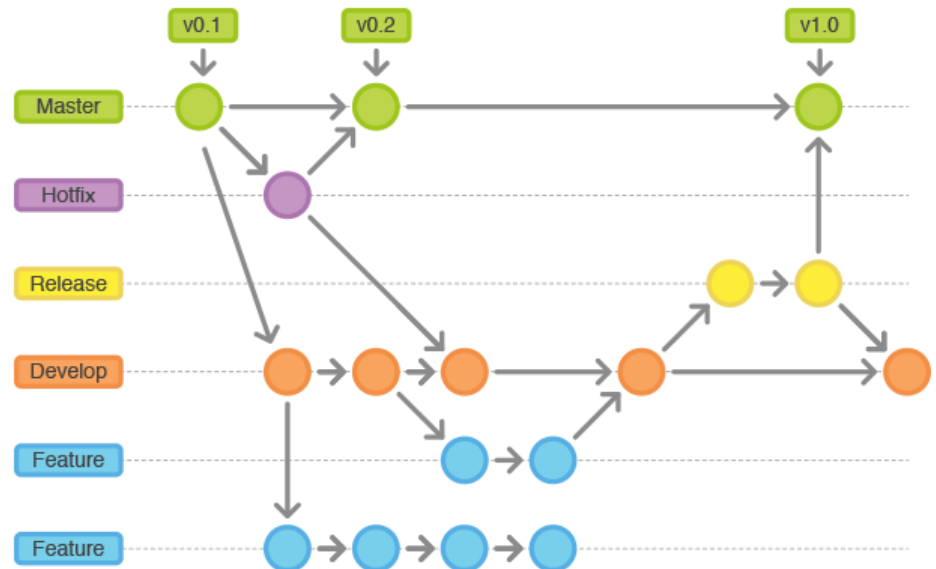


Continuous integration (CI)

- Why continuous integration?
 - Reduce manual, repetitive work
 - Get notified early about issues
 - Forgot to check in a file?
 - Platform differences?
 - Broke previously working functionality?
 - Avoid before-the-deadline chaos
 - Constantly available information about build, constantly up-to-date staging environment

Branching model in VCS

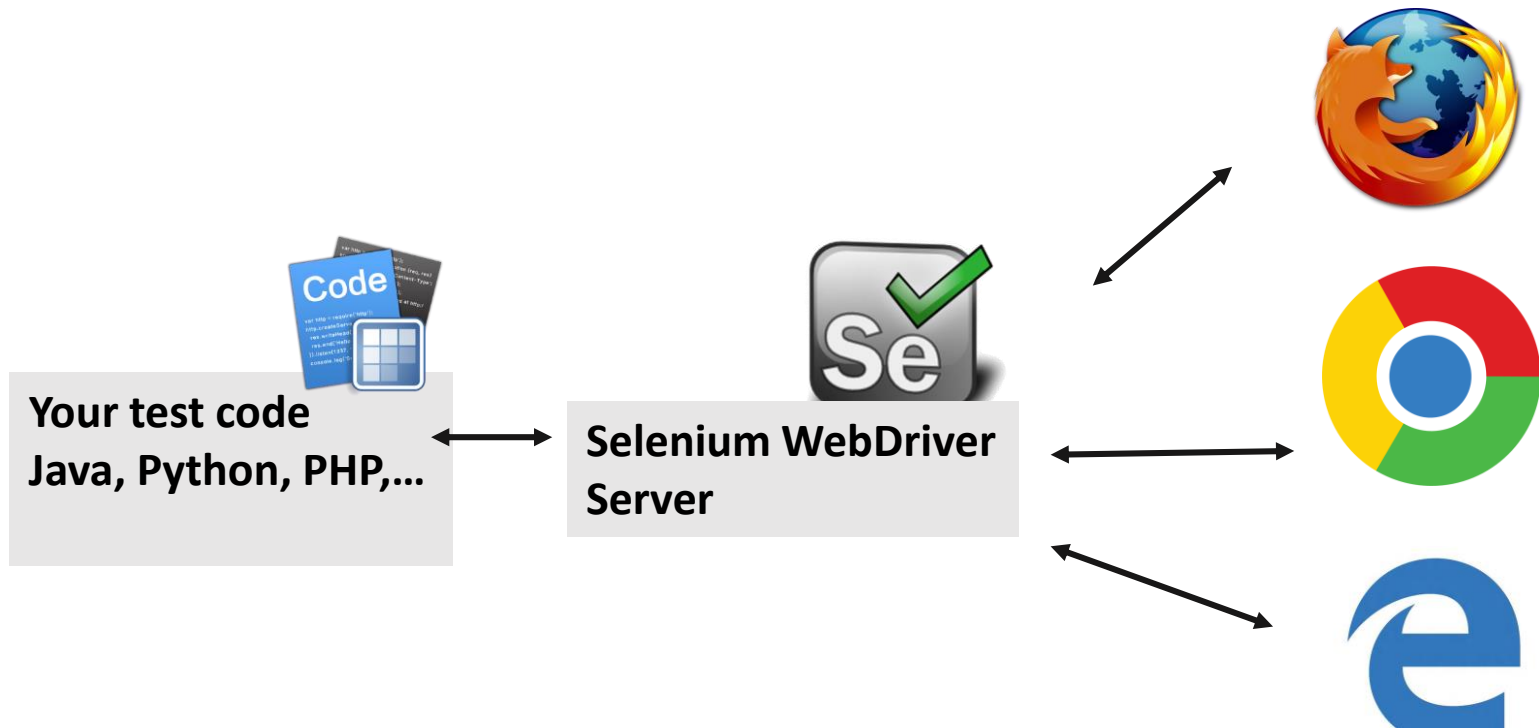
- Everything in master, everybody commits to master
 - Run CI on master
- Feature branches, pull requests to integrate
 - In addition, run CI on every pull request



Automated testing

- Browser based testing
 - Launch application, render in browser
 - **Selenium**
 - PhantomJS
- Integration test, middleware test
 - Test html / JS output from the app
 - Integration testing using mocking

Selenium



How much testing is needed?

- Cover essential use cases with automated tests after they are implemented
- Make sure you test expected and unexpected behavior also

CI Providers

- CircleCI
 - Supports private projects
 - Free 1500 minutes of builds per month
 - Full isolated container, can do whatever you want
- Travis
 - Free for open source, most popular
- Jenkins
 - Host yourself, configure yourself (OpenShift)
- Something else
 - Shippable, drone.io, appveyor



Travis CI



Jenkins

Deployment

- Heroku
- OpenShift
- DigitalOcean (get your Github developer pack)
- AWS

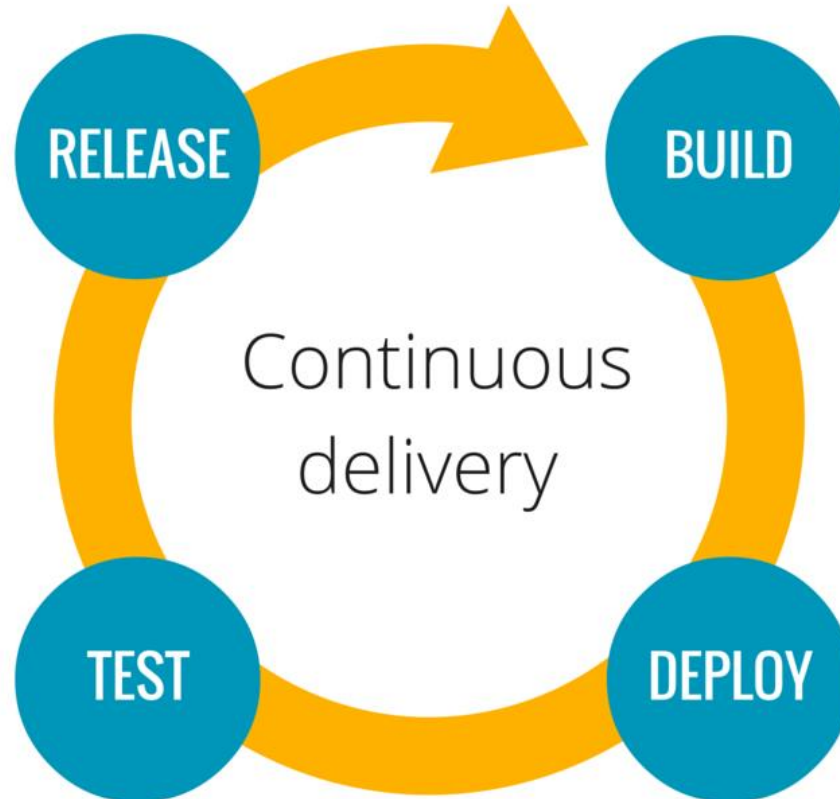
Demo: Example 1

- <https://github.com/riivo/spring-petclinic/>
- PetClinic
- Spring web application in Java
- Scenario
 - Build in CircleCI
 - Run tests, and selenium tests
 - Selenium managed fully through maven
 - If OK, deploy to Heroku

Demo: Example 2

- <https://github.com/riivo/sample-php-mysql-heroku>
- Simple PHP “webpage”
- Setting up database in CircleCI, using database in Heroku
- Run selenium server manually

Aim for continuous delivery



What's next?

- Start setting up CI!
 - It will take time to set up CI, automated deployment, but less time than the manual labor you would spend on it by the end of course!
 - Assign setting up CI to a specific person

- Do not focus on points only, think about benefits
- Make sure you have at least staging environment

- Happy to help you: riivokik@ut.ee

Image Credits

- https://insights.sei.cmu.edu/assets/content/image%20for%20continuous%20integration%20and%20devops_01262015.jpg
- <http://www.accessa.eu/wp-content/uploads/2015/11/Continuous-Delivery-schema.png>