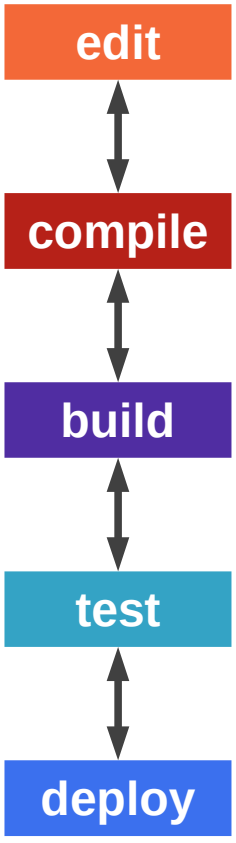
Continuous Delivery

Prof. Dr. Dirk Riehle

Friedrich-Alexander University Erlangen-Nürnberg

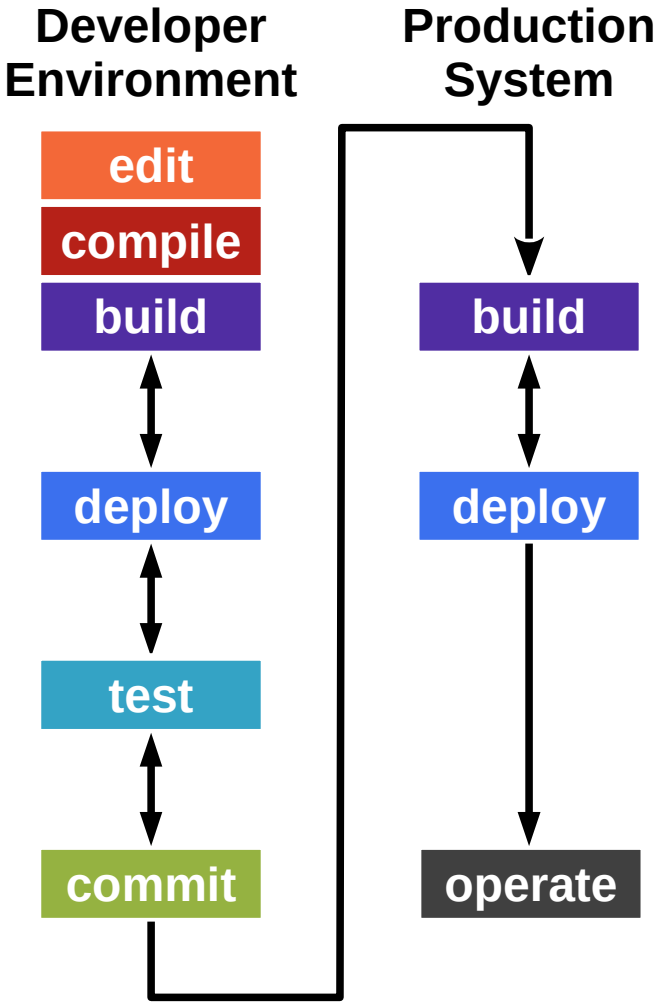
AMOS F02

Beginner's Development Cycle



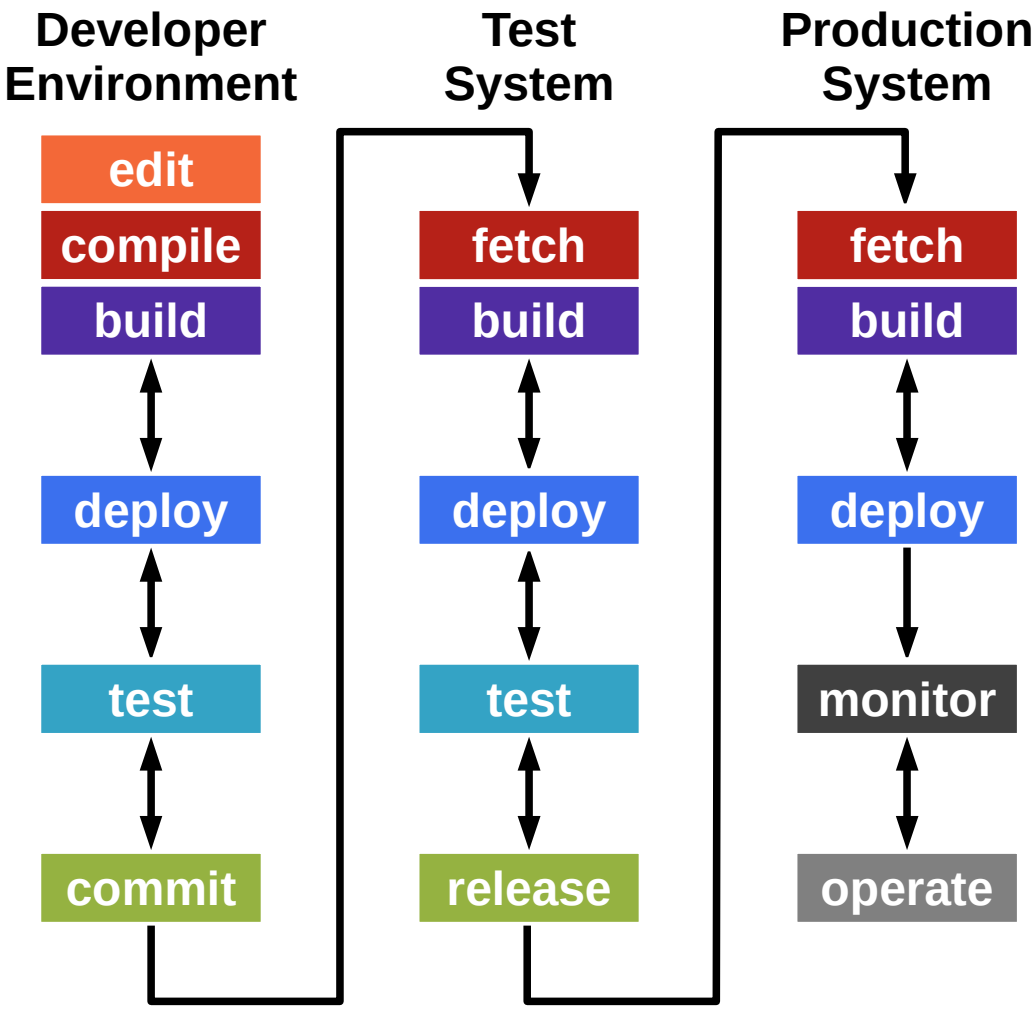
- Developer
  - Edit: Implements new feature
    - Iterates over the code until it looks right
  - Compile: Compiles the code
    - Iterates over the code until it compiles (no syntax error)
  - Build: Puts classes, build path together
    - Packages jar, if any, by hand
  - Test: Tests the program
    - Keeps going until "behavior looks right" i.e. no bugs
  - Deploy: Puts code into production
    - If a student, submits homework

Practitioner's Development Cycle



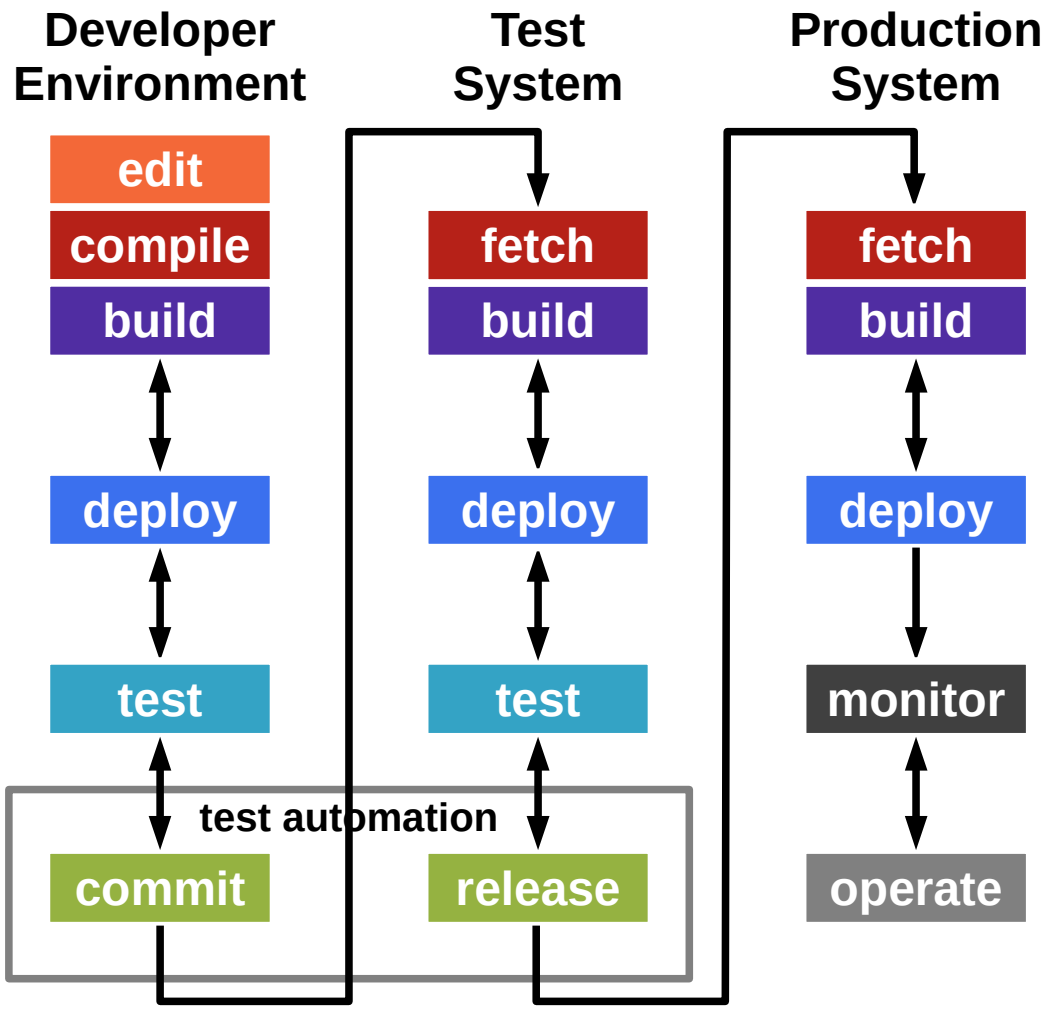
- Developer
  - Finishes development work
    - Uses local edit-compile-test cycle
    - Commits to indicate finishing
  - Builds for production environment
    - Could possibly use local build
  - Deploys to production system

Professional Development Cycle



- QA engineer
  - Fetches code
  - Builds full system
    - Deploys in test system
  - Tests full system
    - Automated and by-hand
      - Component tests
      - Acceptance
      - Integration tests
  - Deploys full system
  - Operates system

Continuous Integration Development Cycle



- Release (QA) engineer
  - Fetches code
  - Builds full system
    - Deploys in test system
  - Tests full system
    - Automated and by-hand
      - Component tests
      - Acceptance
      - Integration tests
  - Deploys full system
  - Operates system

Test Automation

- Test automation ...
  - automatically carries out all available tests
    - Component tests (unit tests)
    - Acceptance tests (functional tests)
    - Integration tests and system tests
  - provides feedback to development and QA

Continuous Integration

- Continuous integration (CI) is a code integration process
  - Upon trigger (commit to official repository)
    - the system under construction is fetched, built, deployed, and tested
    - in a fully automated way (no human intervention)
  - Feedback upon system status is provided to both
    - developers and
    - managers
- The purpose of continuous integration is to
  - always know where you are standing with respect to the project
  - ideally improve quality such that you can deploy at any time
- Continuous integration requires test-driven development

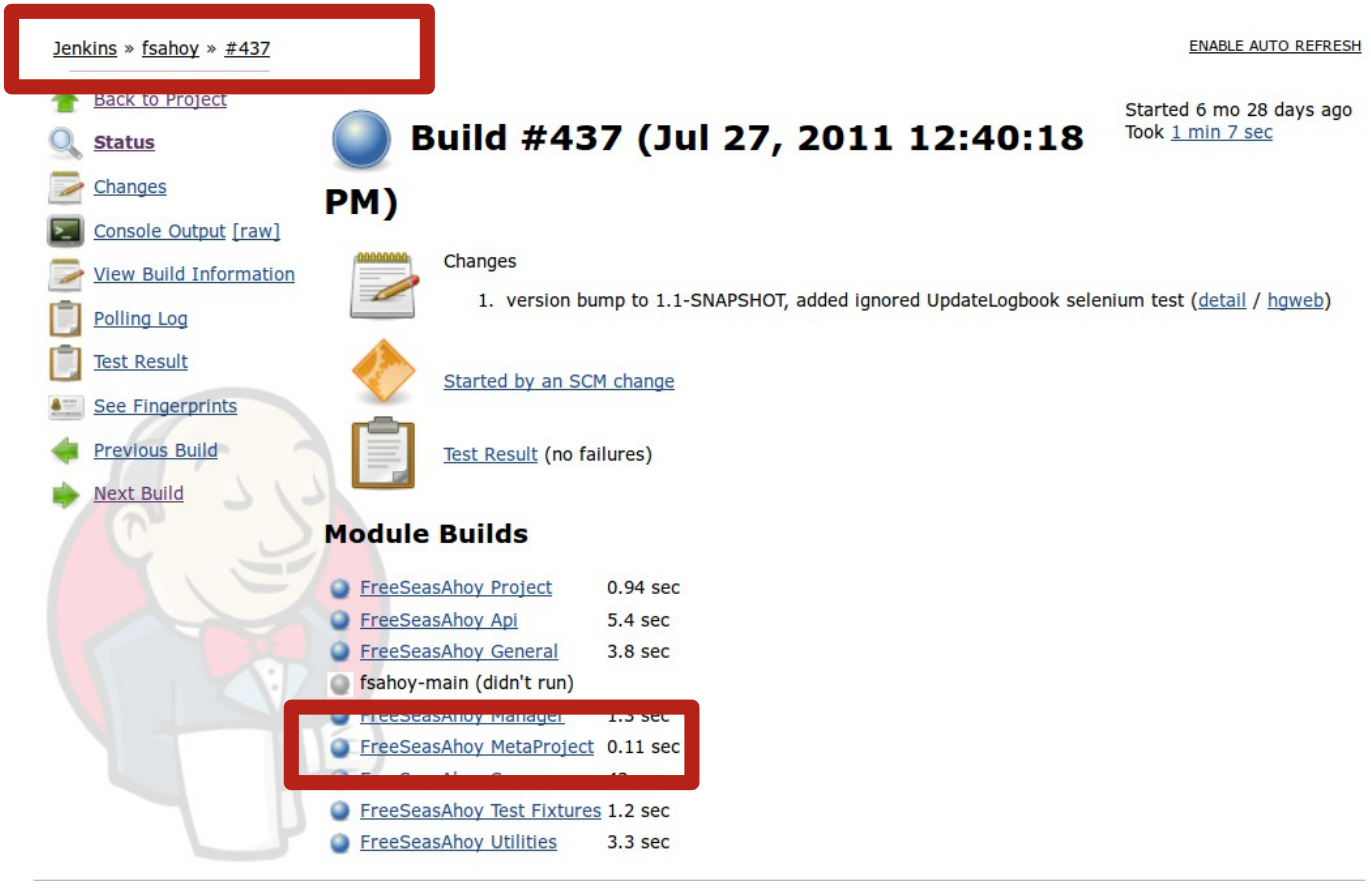
Continuous Integration and Lava Lamps

In the early days, lava lamps were used to signal whether the project could be deployed to production or not.

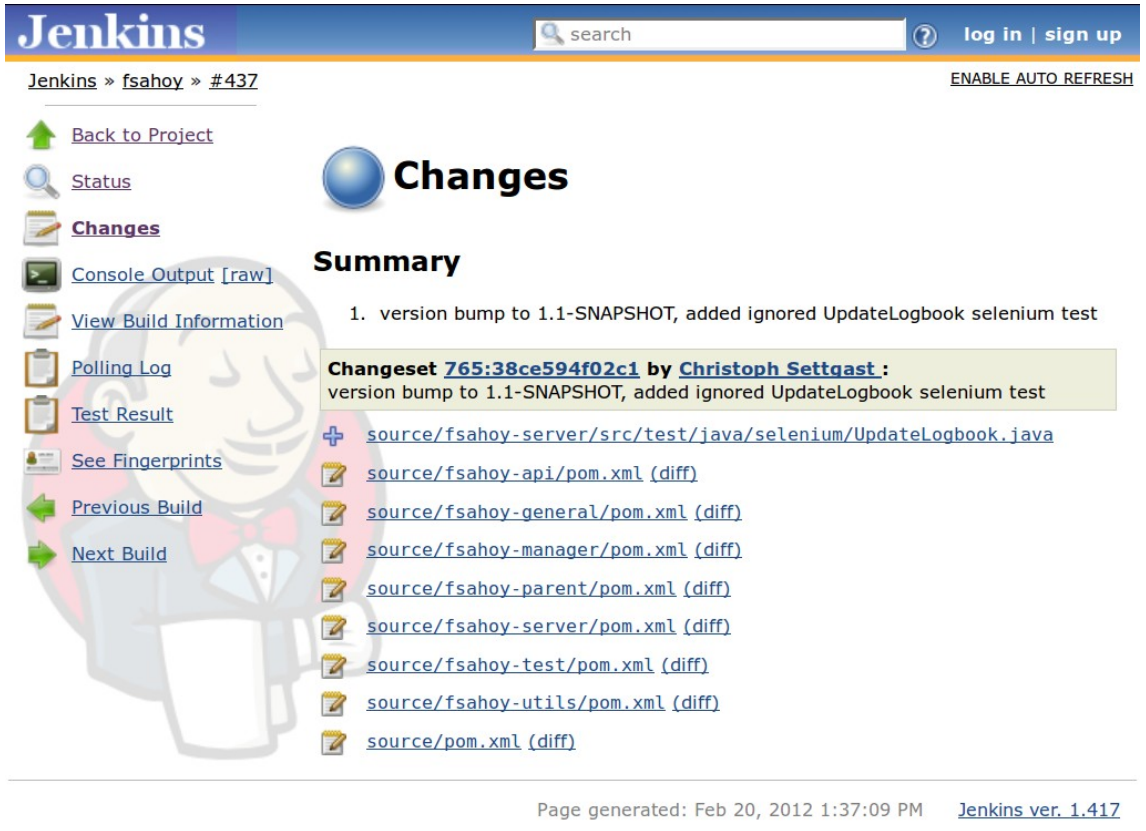
Continuous Integration Dashboards

Continuous integration quickly evolved to build and test status dashboards hung on office walls for everyone to see.

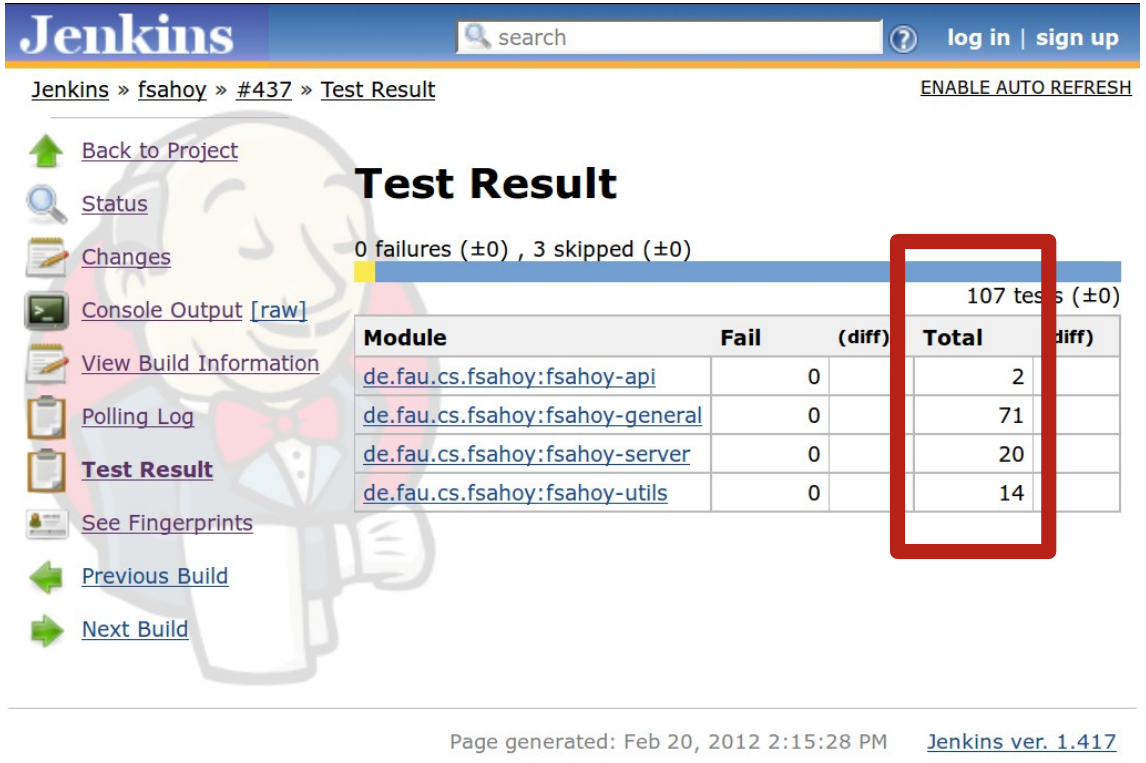
Example Dashboard Build #437



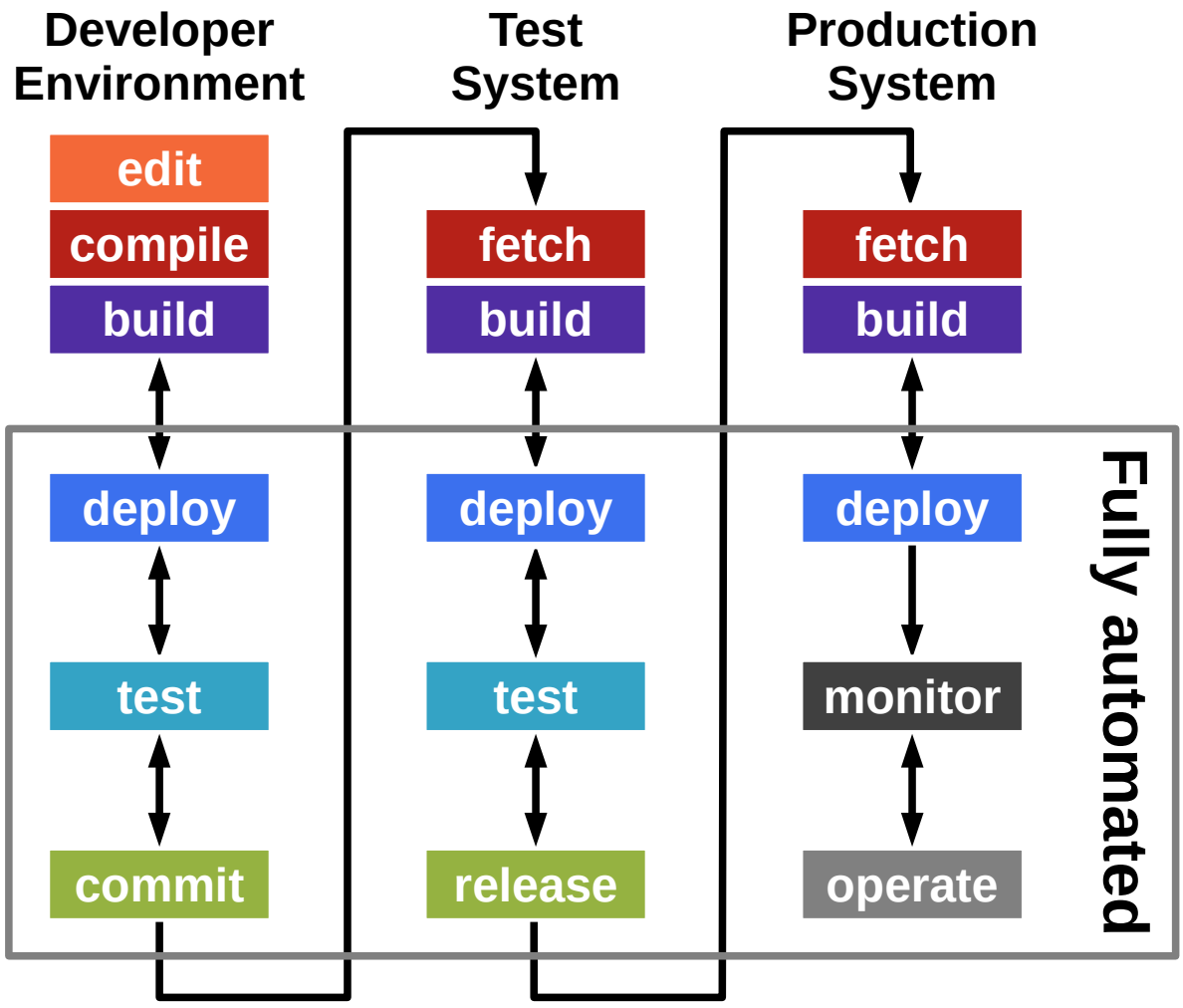
Example Dashboard Changes #437



Example Dashboard Test Results #437



Continuous Deployment Development Cycle



- Fully automated
  - Compile and build
  - Deployment
    - To test environment
    - To production
  - Test execution
- Partially automated
  - System monitoring
  - Automated rollback
- Human decisions
  - Commit decision
- No release decision

Continuous Delivery

- Continuous delivery is a delivery process
  - Upon trigger (commit to official repository)
    - the system is integrated, tested, and **deployed to production**
    - in a fully automated way (no human intervention)
  - A poorly functioning system may be rolled back
    - Requires monitoring and rollback facility of deployed system
    - System status is assessed using key figures
- The purpose of continuous delivery is to
  - put development results into production as fast as possible
  - improve quality by holding the team to high operational standards

Continuous Delivery 1 / 2

1. Test automation
2. Continuous integration
3. Continuous deployment [1]

[1] Short for “development operations”

Continuous Delivery 2 / 2

- Test automation =
  - Tests and testing
- Continuous integration =
  - Test-driven development +
  - Automated building +
  - Test automation
- Continuous deployment =
  - Continuous integration +
  - Deploy to production +
  - Monitoring and rollback
- DevOps
  - Continuous deployment +
  - Operations and culture

Review / Summary of Session

- Continuous delivery
  - Test automation
  - Continuous integration
  - Continuous deployment
  - Development operations

Thank you! Questions?

dirk.riehle@fau.de – http://osr.cs.fau.de

dirk@riehle.org – http://dirkriehle.com – @dirkriehle

Credits and License

- Original version
  - © 2012-2020 Dirk Riehle, some rights reserved
  - Licensed under Creative Commons Attribution 4.0 International License
- Contributions
  - ...