Continuous Integration (CI)
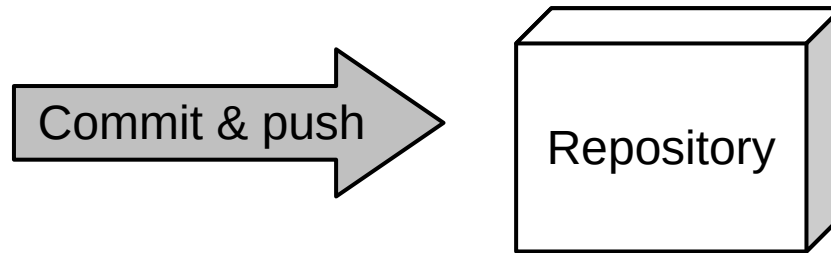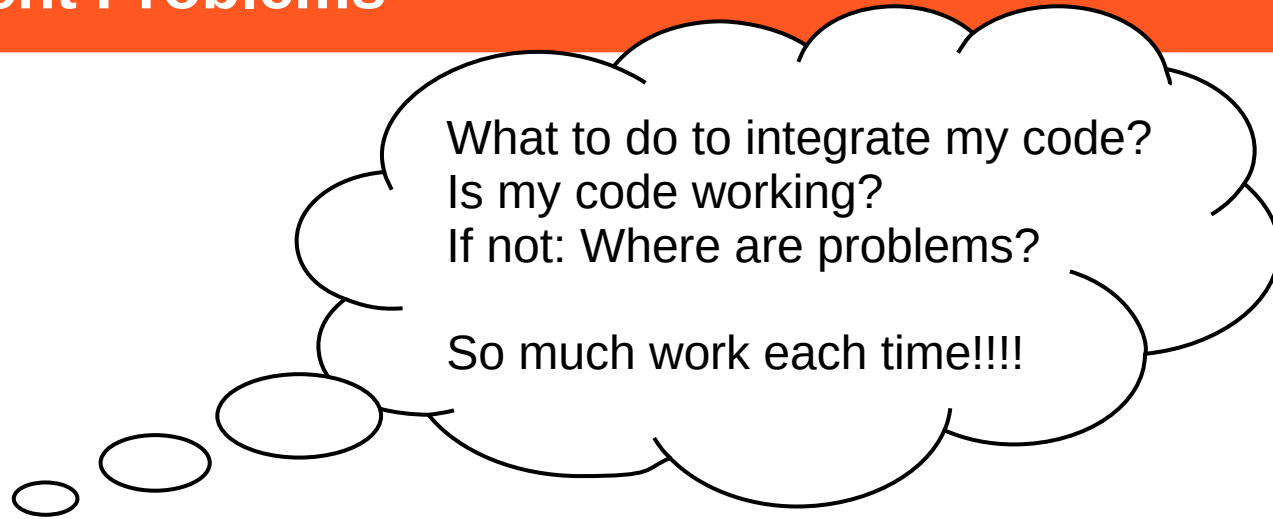
Professorship of Open Source Software

Friedrich-Alexander University Erlangen-Nürnberg

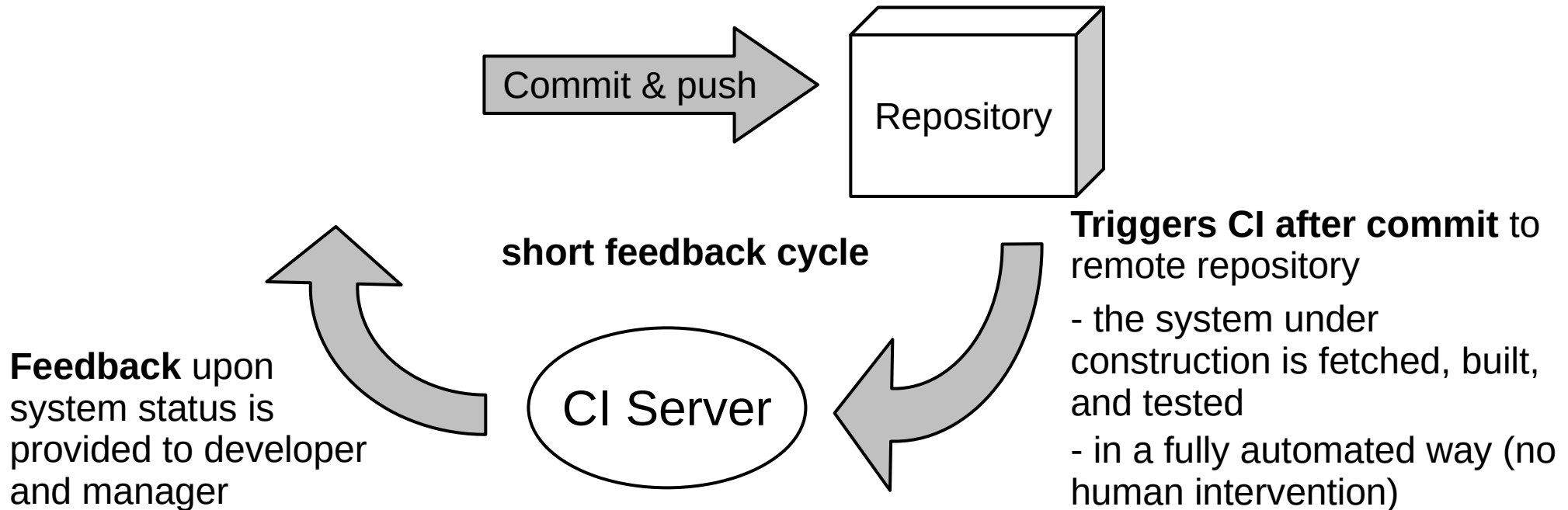**ADAP B02**

What to do to integrate my code?
Is my code working?
If not: Where are problems?

So much work each time!!!!

Commit & push

Repository

Continuous Integration (CI)

"Continuous Integration is a software development practice where members of a team integrate their work frequently […] this approach leads to significantly reduced integration problems and allows a team to develop cohesive software more rapidly." [1]

Commit & push → **Repository**

short feedback cycle

CI Server

Triggers CI after commit to remote repository

- the system under construction is fetched, built, and tested

- in a fully automated way (no human intervention)

Feedback upon system status is provided to developer and manager

[1] Martin Fowler, https://www.martinfowler.com/articles/continuousIntegration.html

Advantages of CI

- Always know if your project is in a healthy state
- Faster integration
- Faster feedback
- Easier to localize bugs
- Frequent integration of the whole application
- Ideally, improve quality such that you can deploy at any time
- Reduce risks introduced by code changes, e.g. refactoring of the application

CI in practice

- Example Tools:
  - Jenkins (https://jenkins.io/)
  - Travis CI (https://travis-ci.org/)
  - GitLab CI (https://about.gitlab.com/product/continuous-integration/)
  - Buddy (https://buddy.works/)
  - ...

https://wiki.jenkins.io/display/
JENKINS/Logo

https://travis-ci.com/logo

https://about.gitlab.com/press/
press-kit/

Advanced Design and Programming
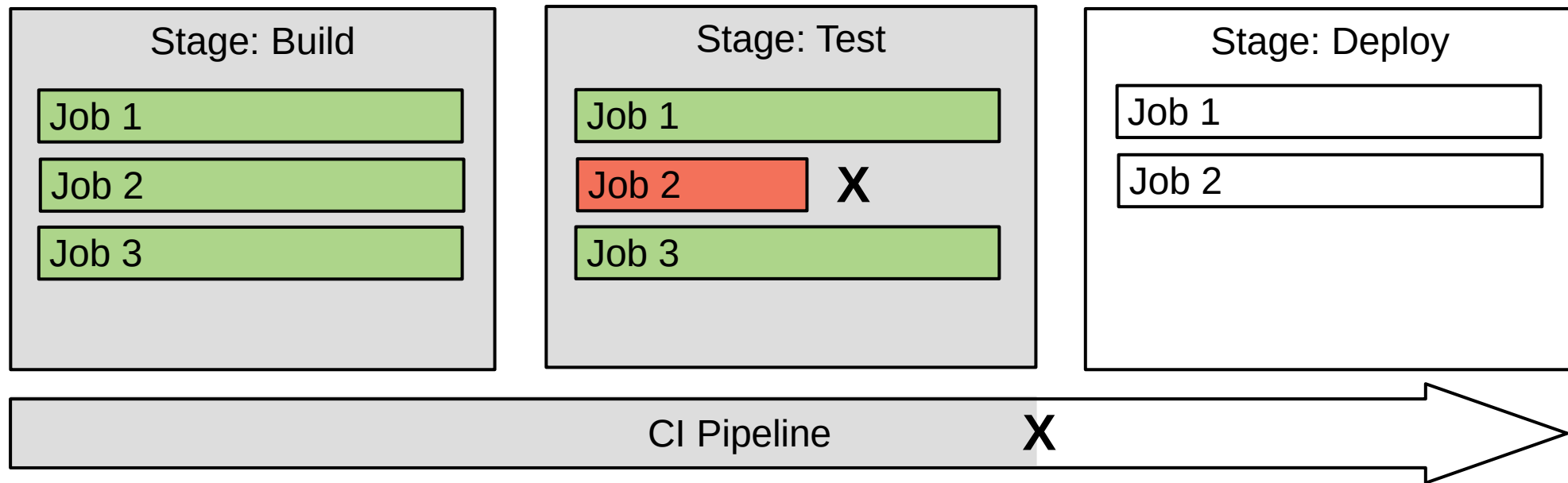© 2021 FAU - Some Rights Reserved

CI Pipeline

- Build Stages – group of parallel jobs, stages are run sequentially
- Jobs – executes a task
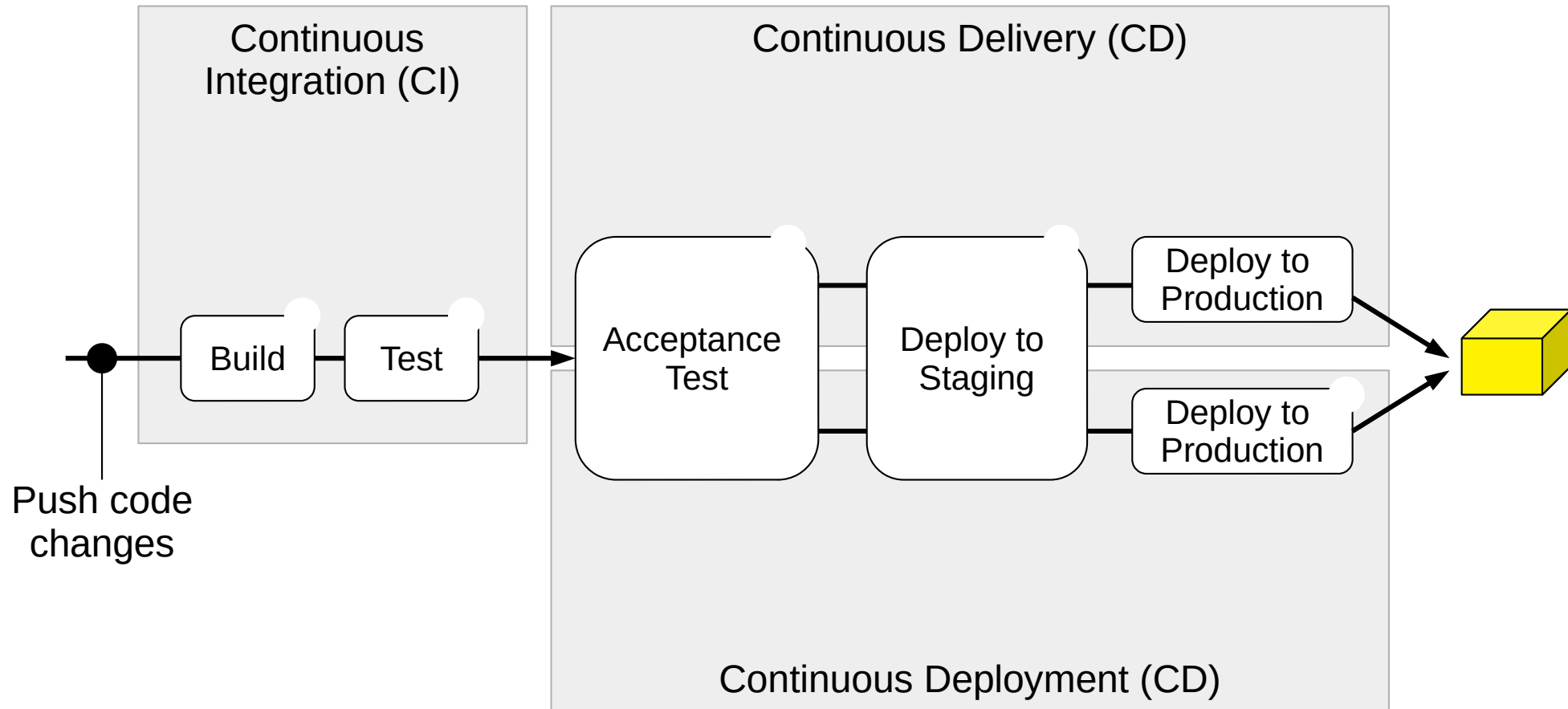- Phases – sequential steps of a job (Job Lifecycle)

Stage: Build	Stage: Test	Stage: Deploy
Job 1	Job 1	Job 1
Job 2	Job 2 X	Job 2
Job 3	Job 3	

CI Pipeline   X →

Advanced Design and Programming

6

Job Lifecycle for Travis CI

.travis.yml

1) OPT Install apt addons
2) OPT Install cache components
3) before_install
4) install
5) before_script
6) script
7) OPT before_cache (for cleaning up cache)
8) after_success or after_failure
9) OPT before_deploy
10) OPT deploy
11) OPT after_deploy
12) after_script

main parts

more about:
https://docs.travis-ci.com/user/job-lifecycle

Extensions of Continuous Integreation

Thank you! Questions?

dirk.riehle@fau.de – **https://oss.cs.fau.de**

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

DR

Credits and License

- Original version
  - © 2019-2021 Friedrich-Alexander University Erlangen-Nürnberg, some rights reserved
  - Licensed under Creative Commons Attribution 4.0 International License

- Contributions
  - Julia Krause (2019)