

Continous.*

Wojciech Barczyński
wojciech.barczyński@wsb.wroclaw.pl

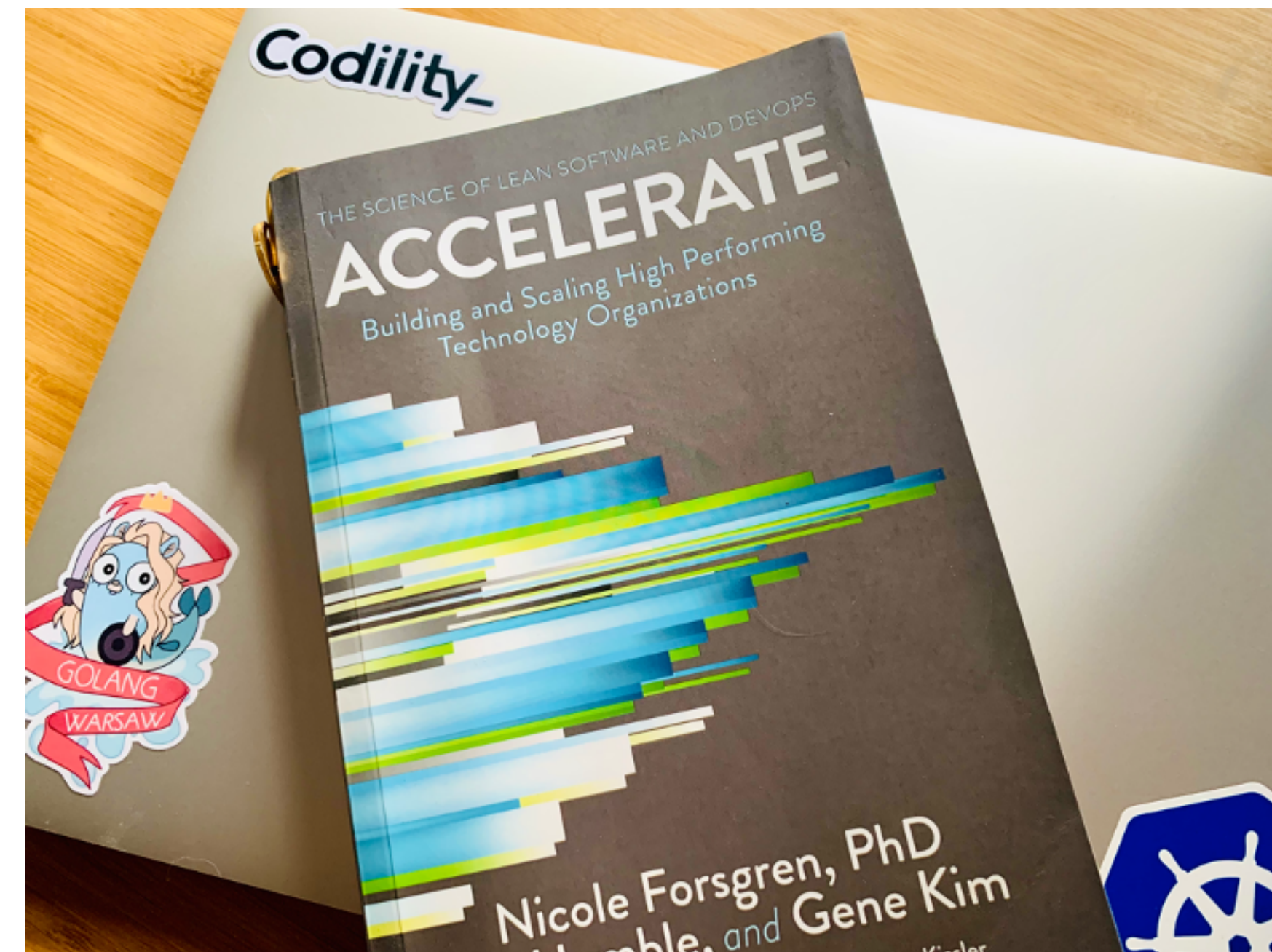
Course materials

github.com/wojciech11/se_software_build_automation_tools

High performance teams

(Tech) metrics:

- Lead Time
- Deployment frequency
- Mean time to Recovery
- Change Fail Percent



High performance teams

Not only technical:

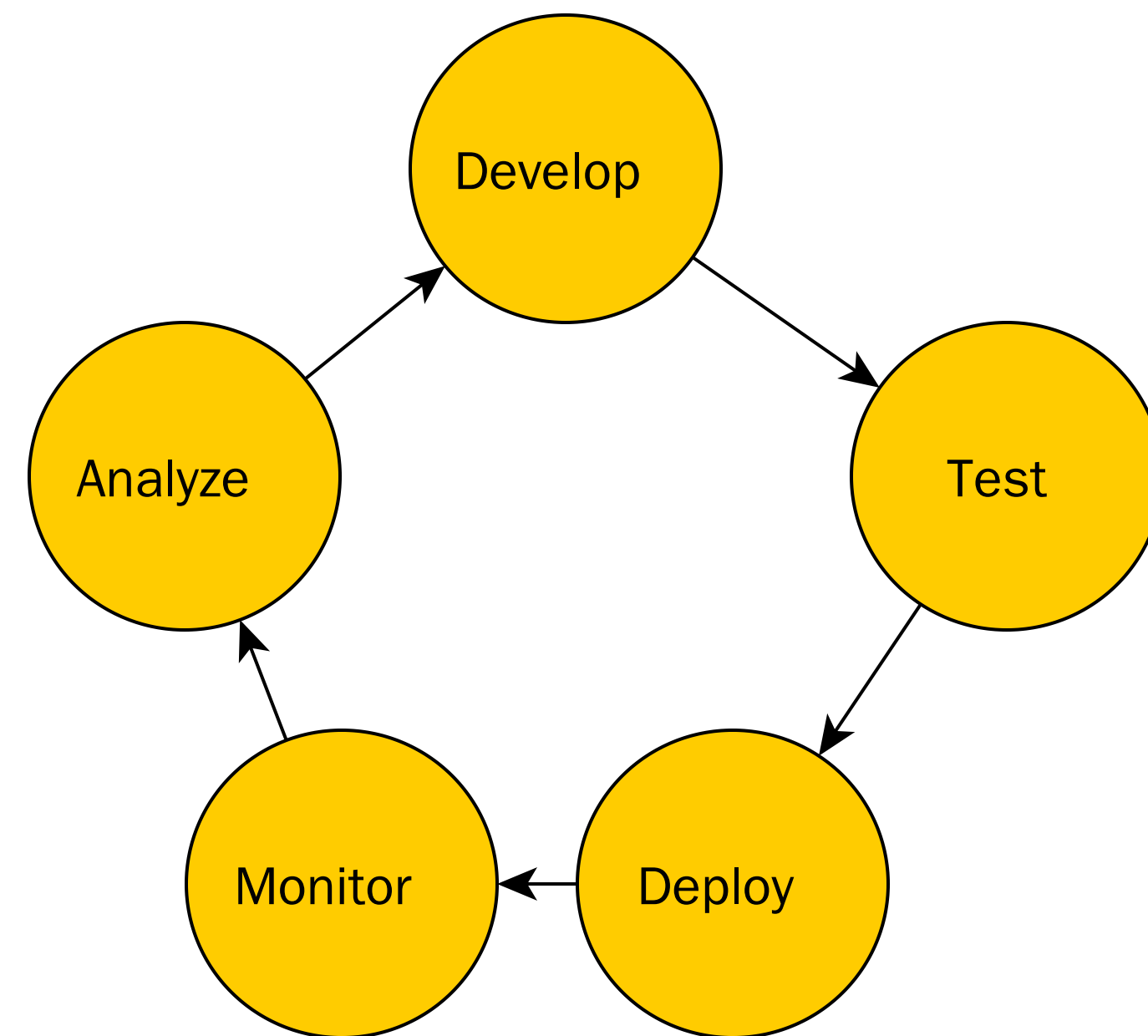
- Trust
- Psychological safety
- High cooperation
- ...

High performance teams

Not only technical:

- Trust
- Psychological safety
- High cooperation
- ...

CONTINUOUS.*



Why

- This is how we build software in the companies

Continuous

- Integration (1)
- Delivery (1)
- Deployment (2)

CONTINUOUS.*

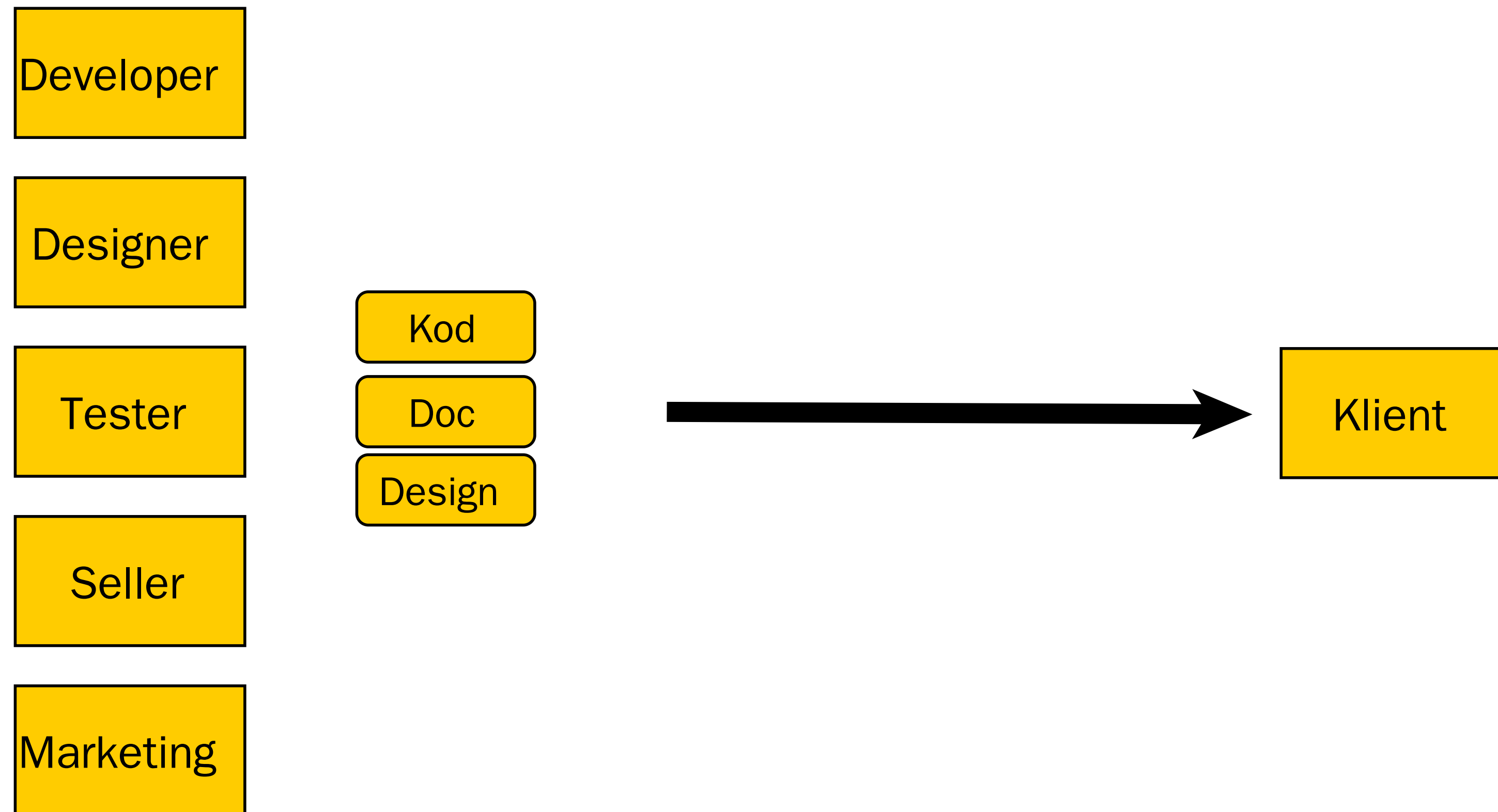


Continuous
Integration

Continuous
Delivery

Continuous
Deployment

CONTINUOUS.*



GOAL

- We know where we are with our code

Practices

- Automatization
- Standarization
- Tooling
- Software design and architecture
- Cooperation between teams and people

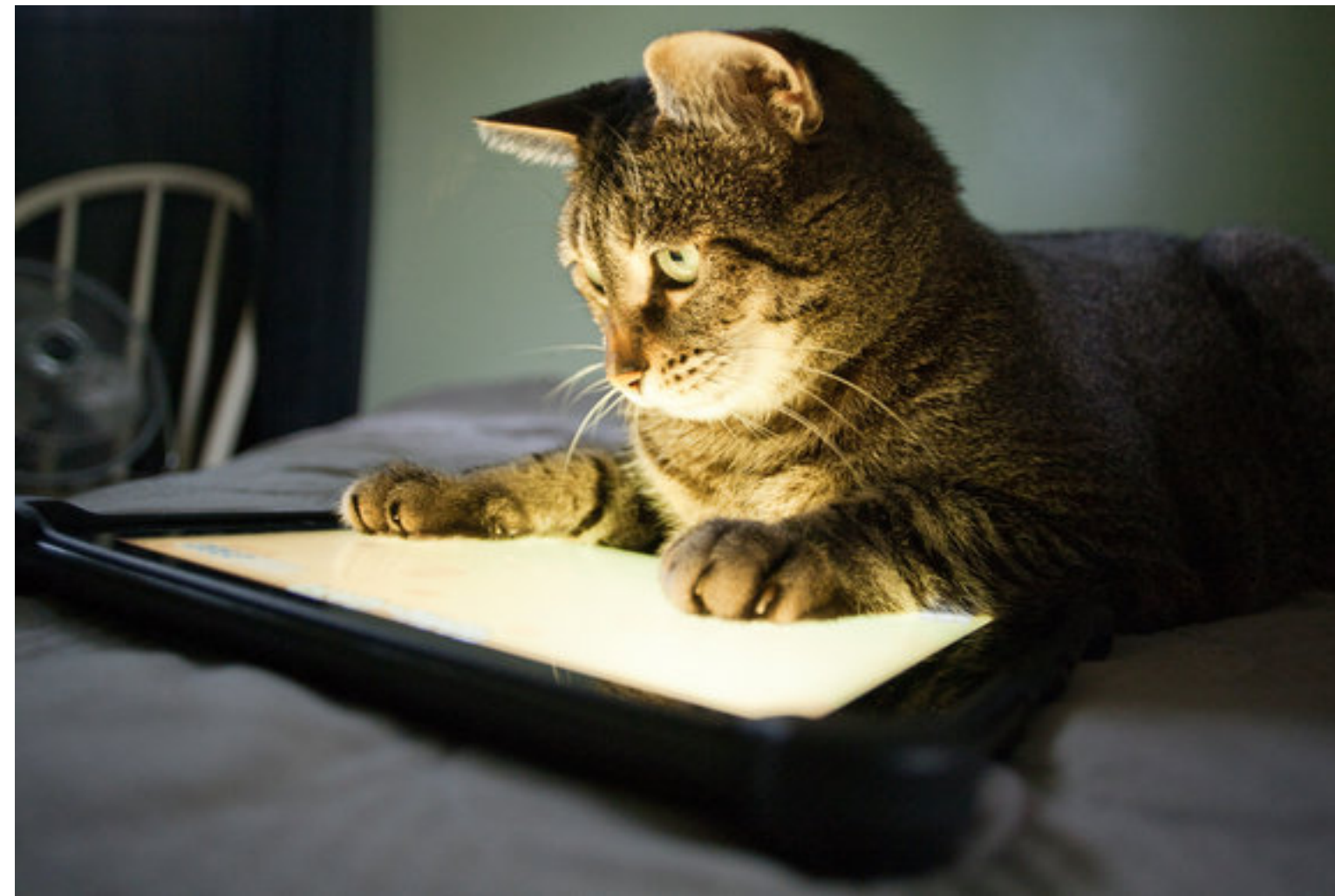
Continuous integration

- Every change validated
- We can package the software

Continuous delivery

- Our software ready to be installed

Continuous deployment



Our software is in the hands of our customers

Tools

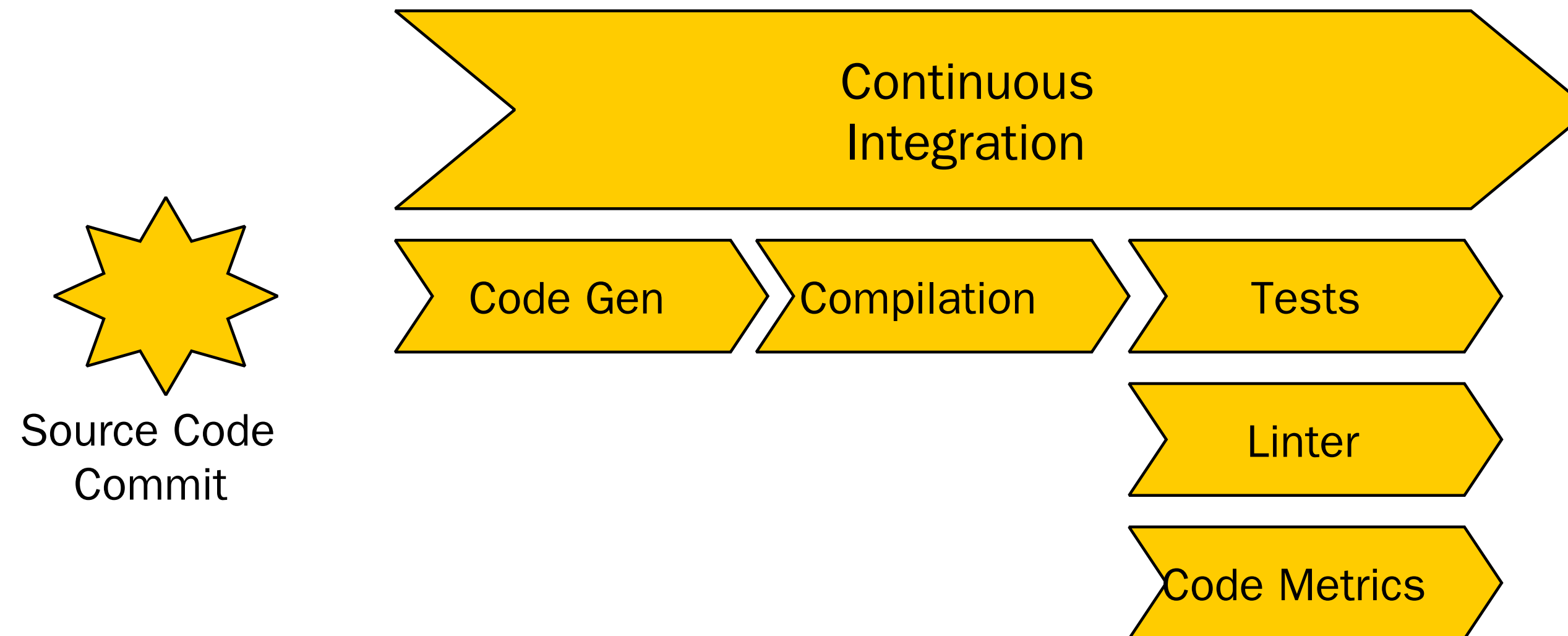
Automation tools:

- Jenkins / GitlabCI
- Github Actions
- TravisCI and many more

Continuous integration



Continuous integration



**Working CI/CD has the highest
priority**

CI starts on the dev workstation

CI starts on the dev workstation

- tool common for the tools and people
- you can run the software locally
- you can run the test locally