Software Architecture – Why I Do It Differently

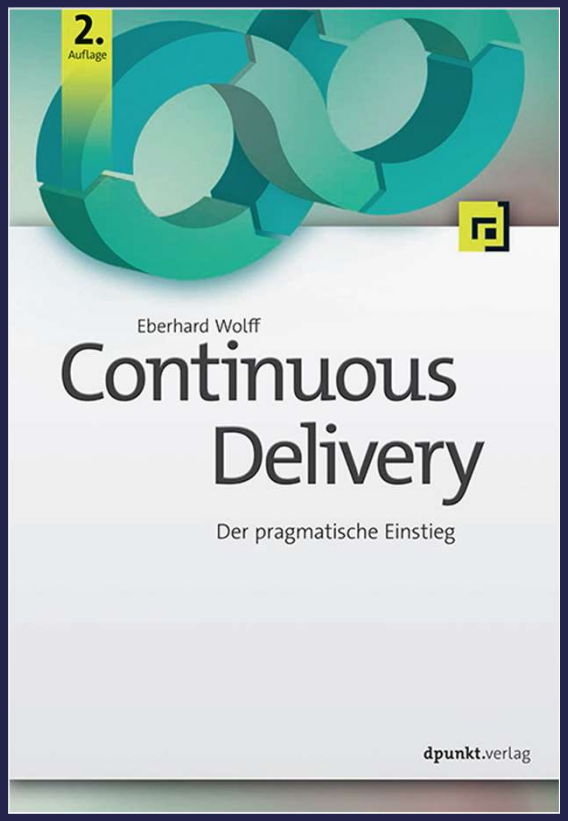
INNOQ

EBERHARD WOLFF

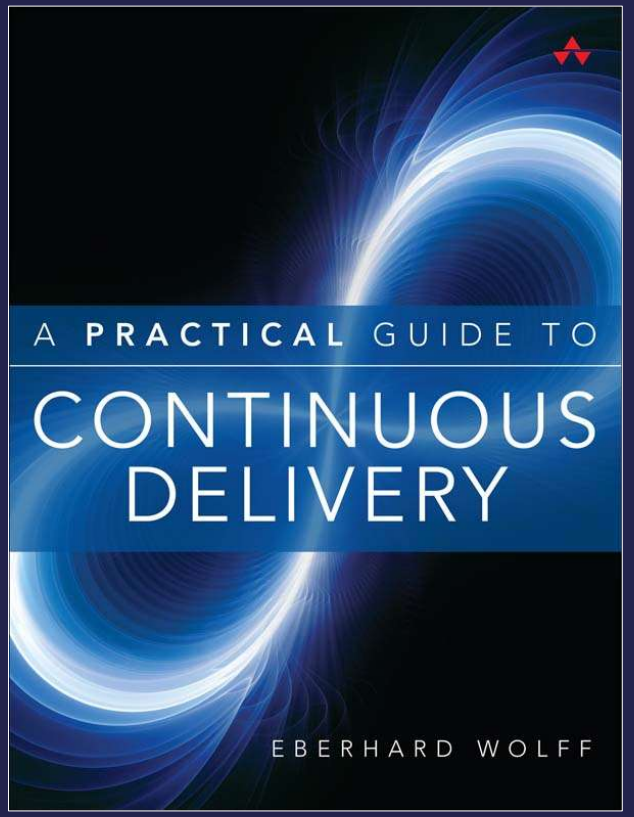
Fellow at INNOQ Deutschland GmbH

@ewolff

www.ewolff.com

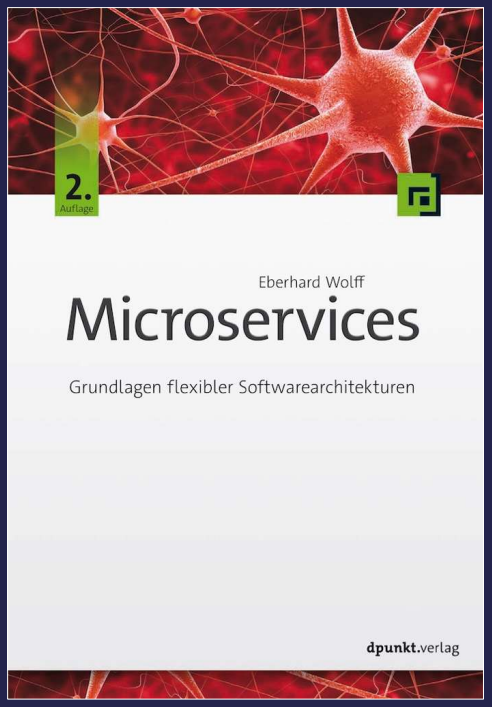


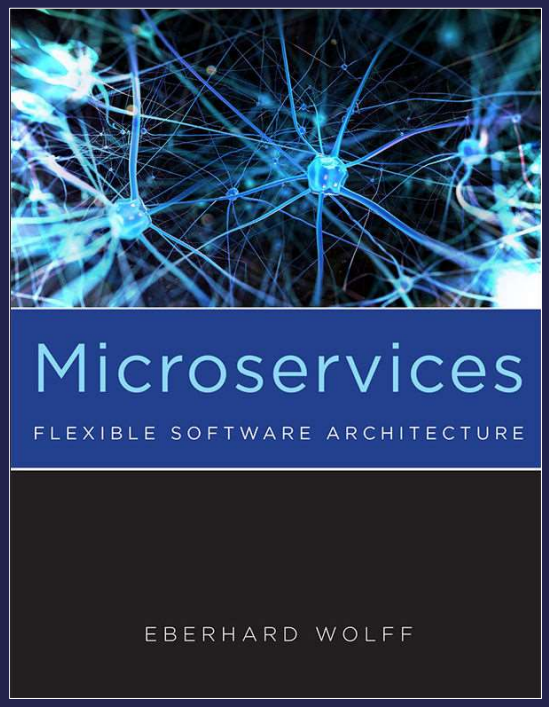
www.continuous-delivery-buch.de

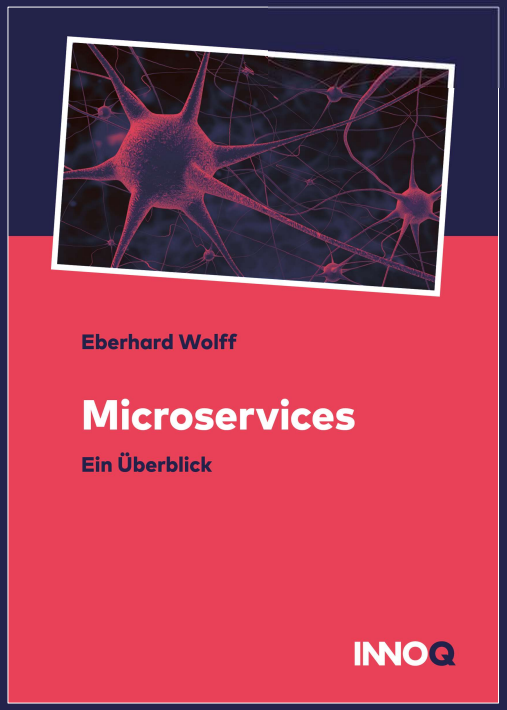


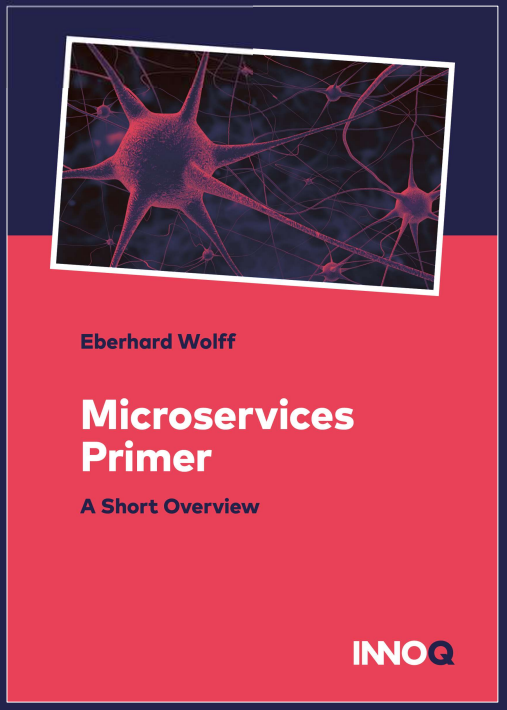
www.continuous-delivery-buch.de

FREE









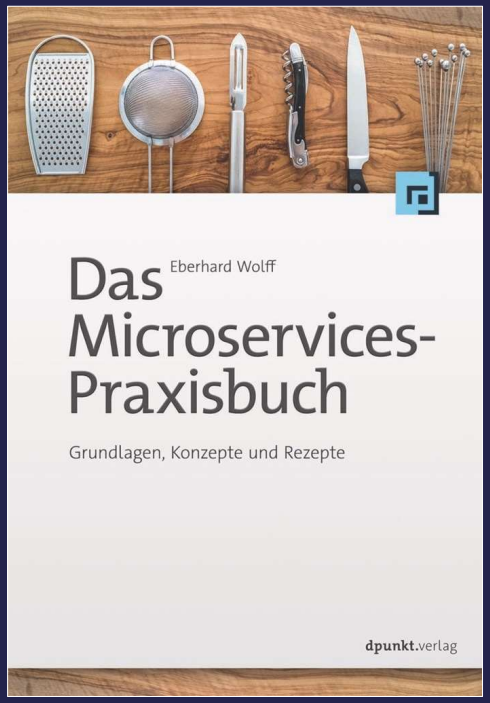
microservices-buch.de

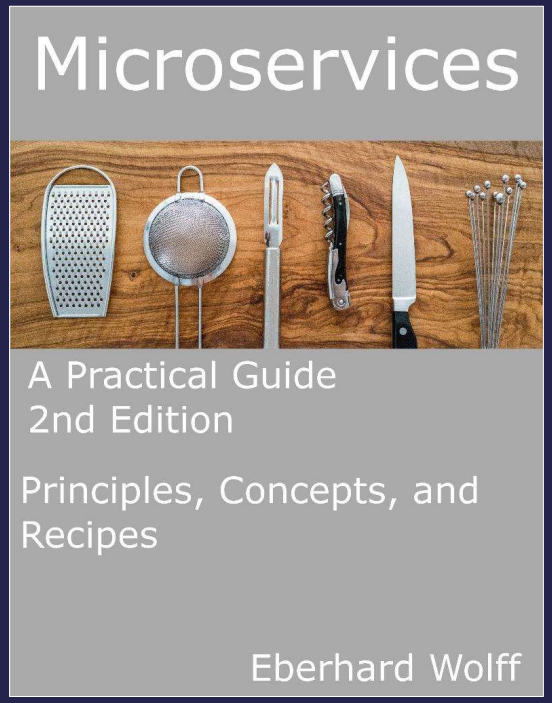
microservices-book.com

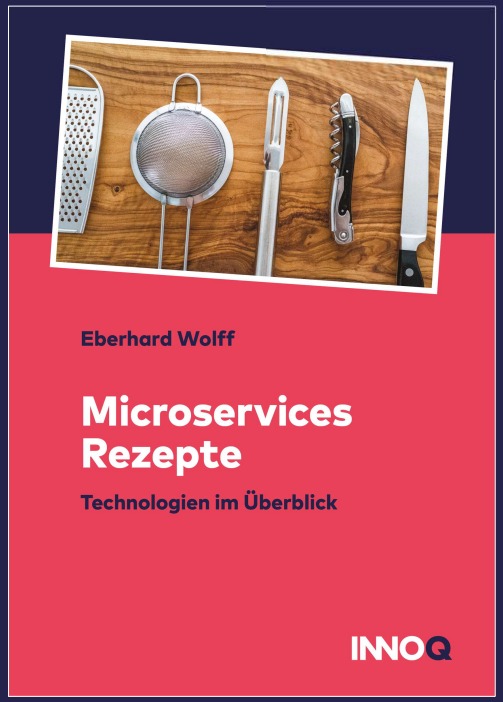
microservices-buch.de/ueberblick.html

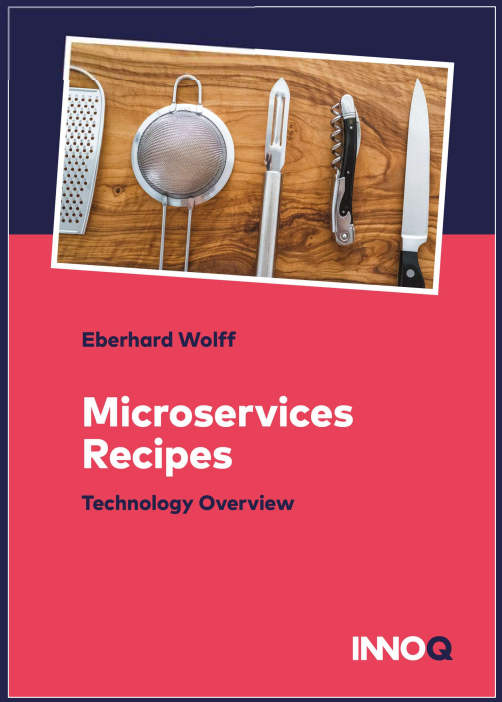
microservices-book.com/primer.html

FREE









microservices-praxisbuch.de

practical-microservices.com

microservices-praxisbuch.de/rezepte.html

practical-microservices.com/recipes.html

https://www.isaqb.org/

Great software architecture: clean, scalable, maintainable

Architecture Fail?

- Software doesn't go into production

  Security problem

  Compliance problem

- Fail caused by structure?
- Successful architecture?

Martin Fowler

- Software architecture =

Important

and hard to change decisions

- How to know in advance?

Photo: Webysther Nunes

Software Architecture

- Find technical solutions

...to the problem at hand.

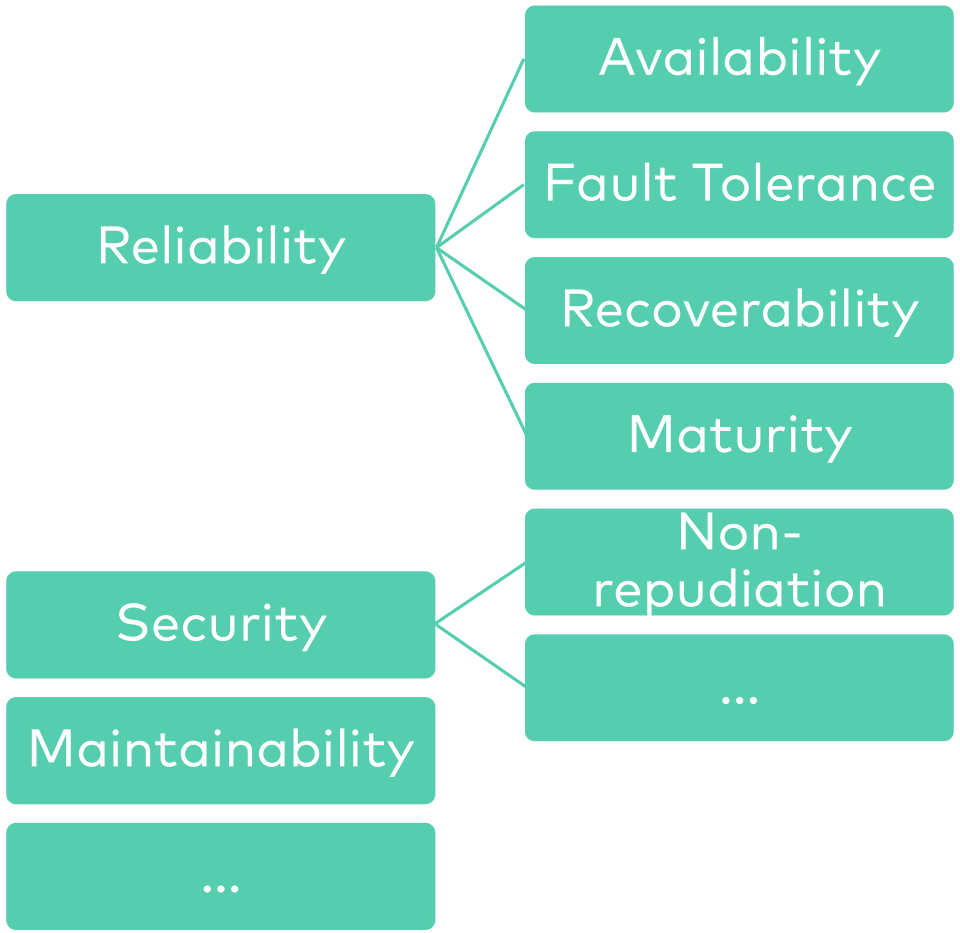
- Home-grown definition
- Broad definition

- Need to understand the problem!

Understand the Problem

Quality Attributes / Tree

ISO 25010



Structure only
helps with
maintainability

Quality Attributes

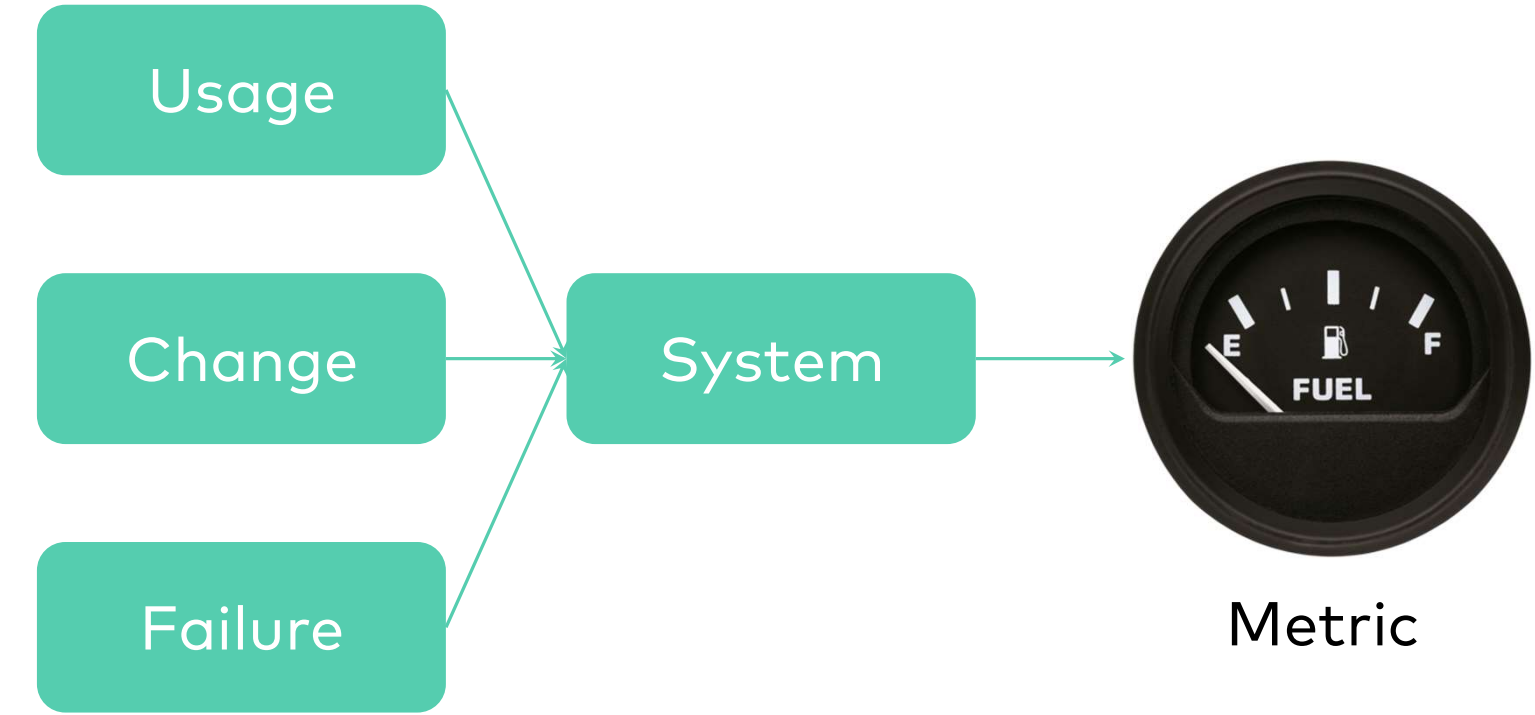
- Holistic view on quality
- Identify important attributes

- But

high-level

hard to verify

Quality Scenario



Quality Scenario

- Concrete

- Easy to verify - metric

Usage Scenario

- Stimulus

  A new users registers

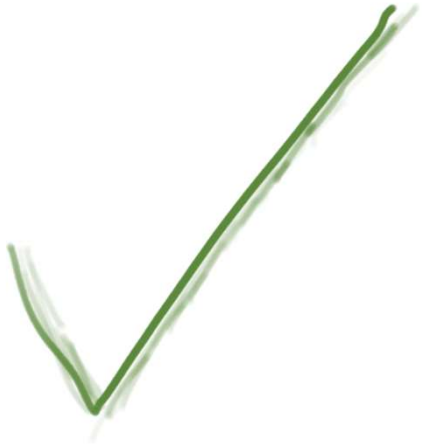
- Metric

  Only one in 1.000 users calls the hotline.

  Usability – Ease of Use

Usage: Solution

- Hire UX experts
- Usability tests



- No "classic" architecture work

Change Scenario

- Stimulus

  A new language / locale should be support

- Metric

  No code modification needed.

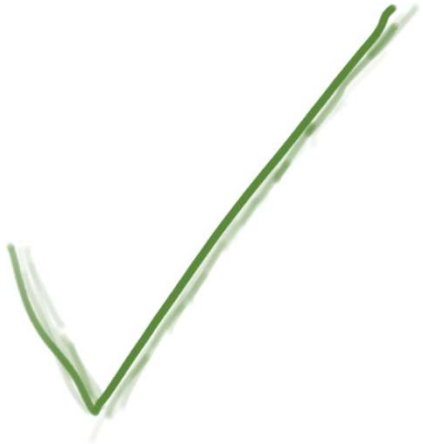
  Takes two days

  Maintainability - Modifiability

Change Solution

- Configuration files for language

- Code quality irrelevant



Failure Scenario

- Stimulus

  A server crashes

- Metric

  System might be unavailable for two hours.

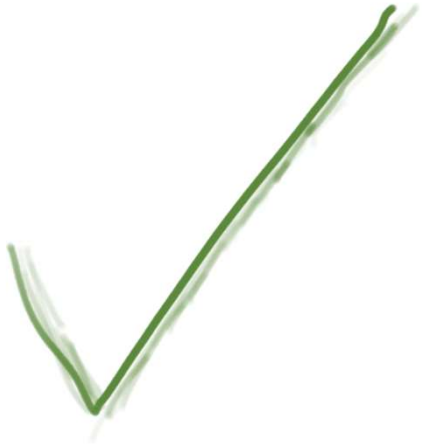
  No data might be lost.

  Reliability – fault tolerance

Failure Solution

- RAID
- Backup
- Data center in different locations

- No need for a cluster of servers



Solutions

- Solutions must solve problems.
- Traditional measures like

  high code quality,

  clean architecture,

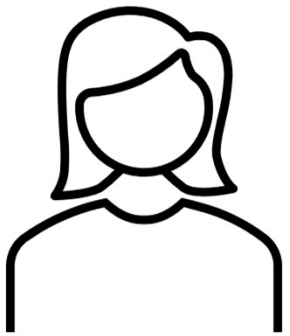
  scalability,

  your favorite framework or language

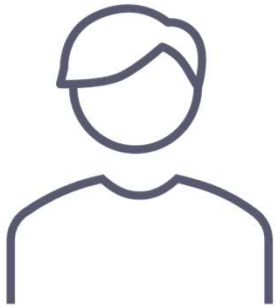
...solves none of the scenarios

How can Architects Do Architecture?

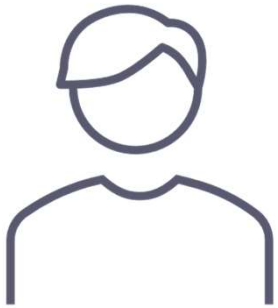
Traditional Architecture



- Let the architect decide everything!
- Architect will be overloaded
- Architect cannot possibly know all details.



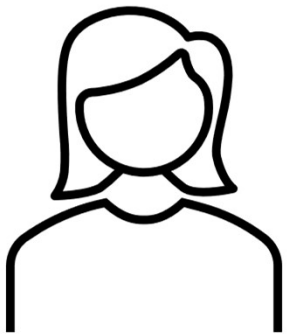
Scrum Master
Removes obstacles
Enforces rules



Product
Owner

Team
Self-organizing
Implements stories

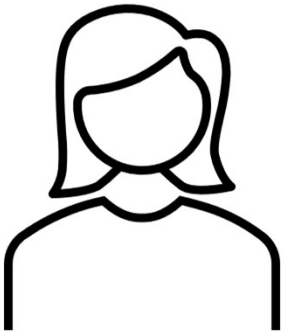
Modern Organization



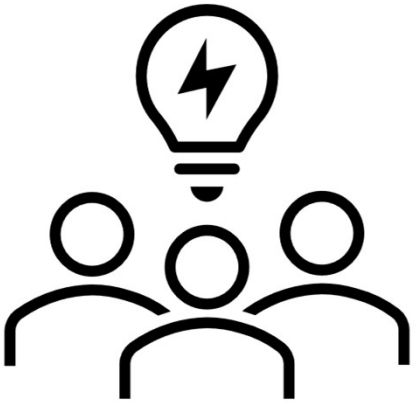
- Self-organization
- No command & control

- Architect can't decide & command

Modern Organization



- Architect can't decide & command.
- How can you even do architecture?



Must enable others

Team = experts

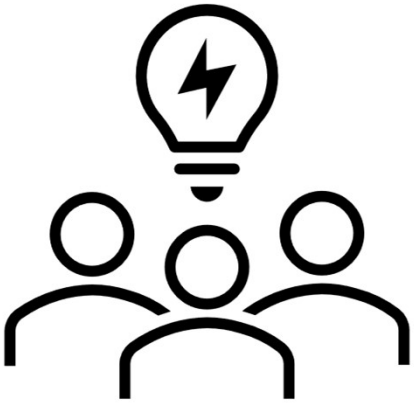
Knowledge about details of the system

Knowledge about technologies

Software Architecture = Collaborative Game

- All lose or win together
- Everyone has a specific role
- Communication is essential

How can architects even influence the architecture?

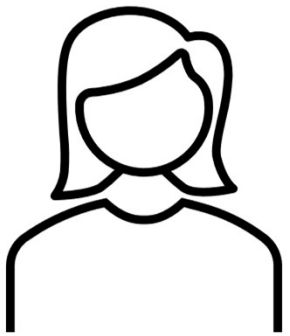


No command & control

Must convince others!

...or come up with better ideas together.

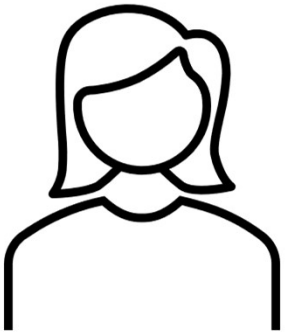
Architect



- So I want to introduce microservices domain-driven design ...

- How?

Architect



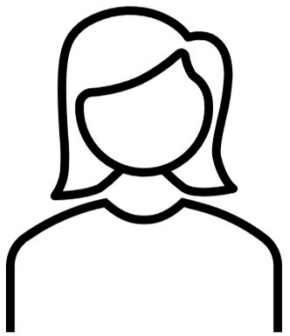
- I can just do it!
- Here is my kewl architecture!

- Makes little sense
- Others must join in

Architects shouldn't do architecture!

Start: More than Trainings

Architect



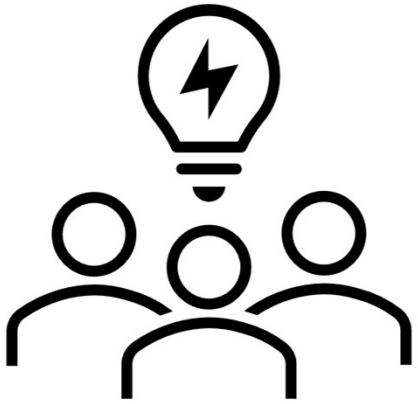
- Spread knowledge about

microservices

domain-driven design

…

- Make them use the knowledge!



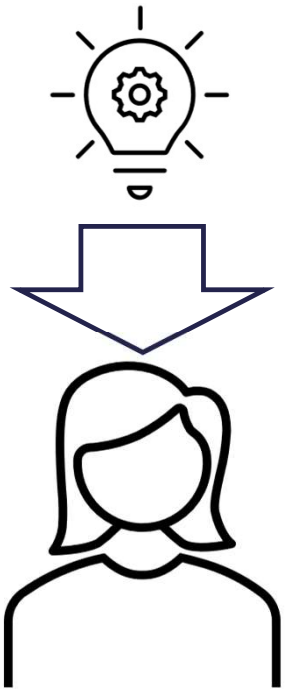
Invite diverse roles

Architecture: many stakeholders

Dev, Ops, QA, Management

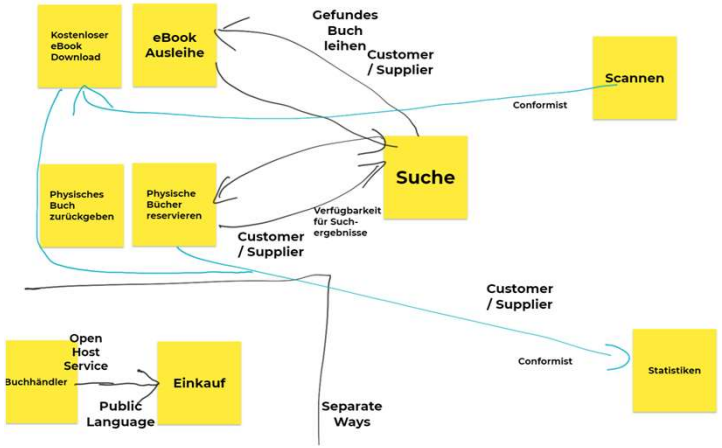
Traditional Training

- Provide knowledge
- Knowledge alone is not enough



Training by Doing

- 5 slides for 5h training
- Not a lot of content
- Rest: doing labs



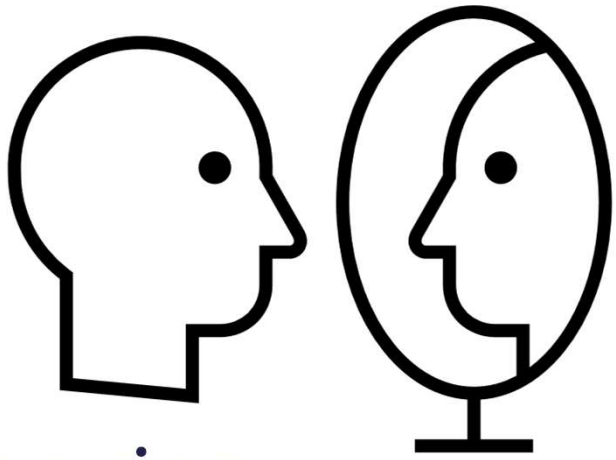
- Afterwards: Attendees have “done” it

…more likely to do it in the future, too!

Training: Reflection

- End of training
- How will you use this on Monday?
- What obstacles do you see?
- Collect post its

...and / or discuss



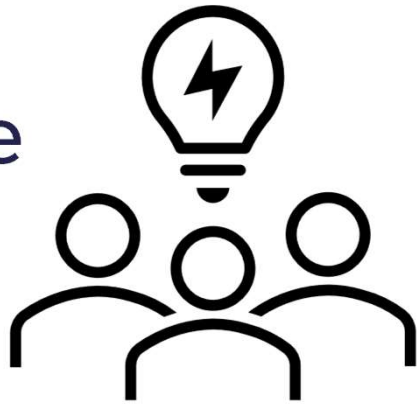
- i.e. think about changing behavior

Design Training Collaboratively

- Group of trainers
- Spreads ideas even better
- To teach, you have to really understand
- Customize trainings and labs

...so they fit the needs even better

- Trainings can continue after people leave

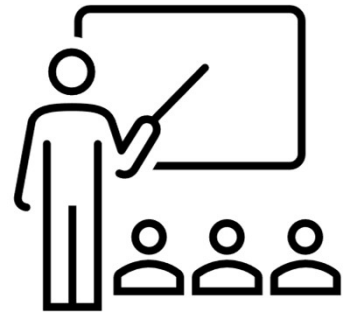


Support

Support

- Training provides basic knowledge

...and people will try the new concepts



- They will run into challenges
- Provide consulting / support

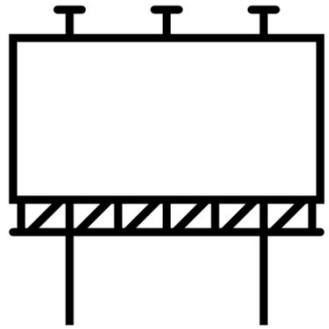
- Why should a single training be enough?

Workshop

Workshop

- Someone has a challenge

e.g. how to design a part of the system

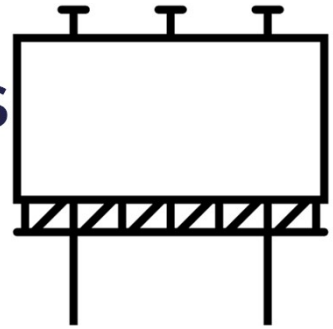


- Make it an exercise!
- Write down the challenge
- Have multiple groups work on it

Workshop: Benefit

- Truly apply new techniques
- Spread knowledge about new techniques

- Spread knowledge about challenges



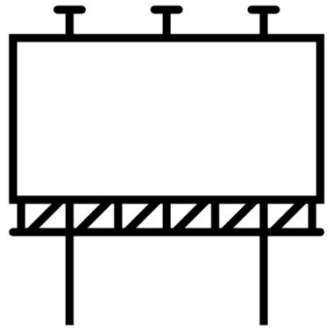
...and decisions

- Strengthen collaboration

Workshop: Challenges

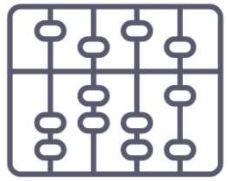
- Effort to prepare

- Instead: Open Space?
- More ad hoc workshops?



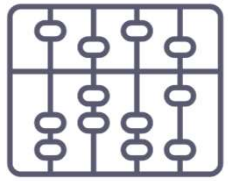
Conclusion

Conclusion



- Software architecture = solve technical problems
- Quality attributes / tree / scenarios to understand problem
- Often solution is not traditional architecture

Conclusion



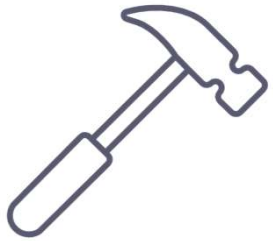
- With self-organization, architects can't just do architecture

Therefore: architects shouldn't do architecture

- ...but must empower
- ...convince
- ...or find other solutions

Architecture Tools

- More than trainings
- Support
- Workshop
- More coaching



Send email to jlove2020@ewolff.com

Link to Dropbox

- Slides
- Service Mesh Primer EN
- Microservices Primer DE / EN
- Microservices Recipes DE / EN
- Sample Microservices Book DE / EN
- Sample Practical Microservices DE/EN
- Sample of Continuous Delivery Book DE

Powered by Amazon Lambda & Microservices

- Email address logged for 14 days, wrongly addressed emails handled manually