



Organization: A Tool for Software Architects

INNOQ

Eberhard Wolff
Fellow

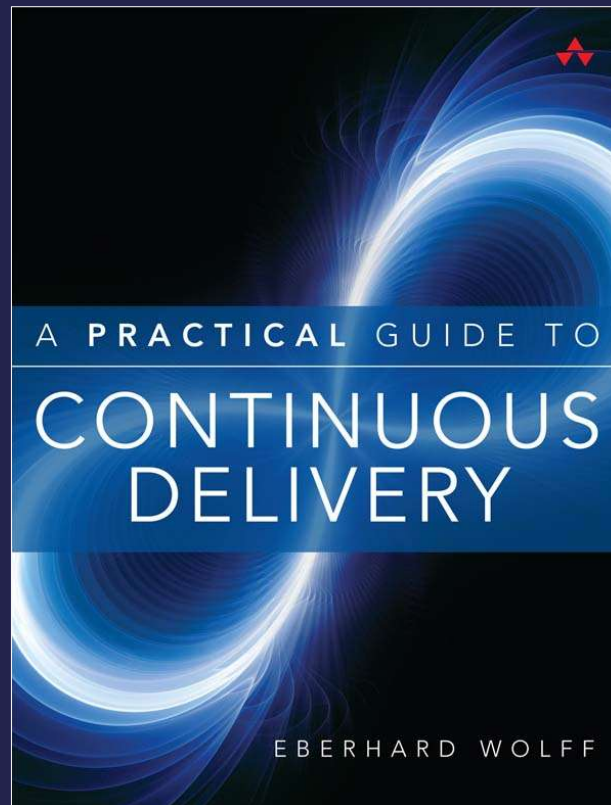
EBERHARD WOLFF

Fellow at INNOQ Deutschland GmbH

@ewolff

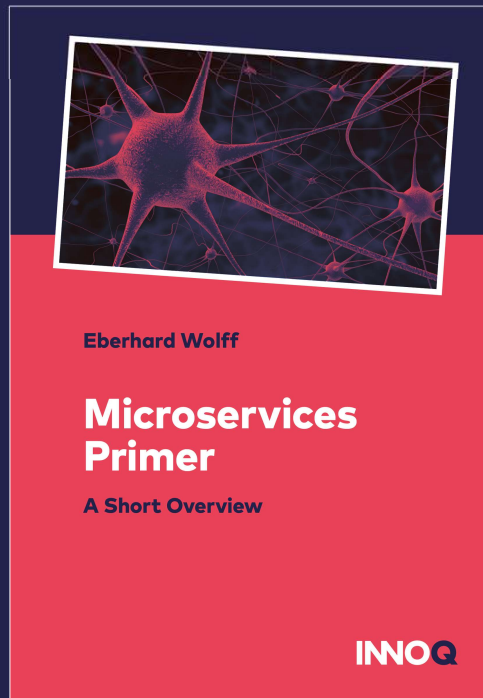
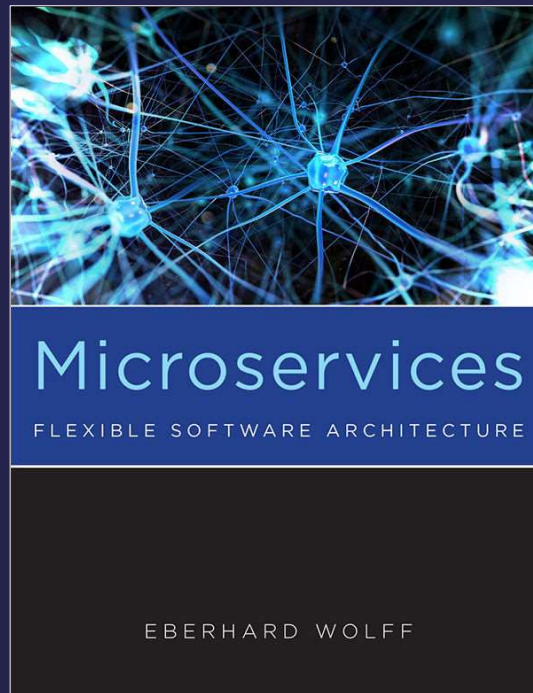
www.ewolff.com





www.continuous-delivery-buch.de

FREE



microservices-book.com

[microservices-book.com/
primer.html](https://microservices-book.com/primer.html)

FREE

Microservices



A Practical Guide
2nd Edition

Principles, Concepts, and
Recipes

Eberhard Wolff



Eberhard Wolff

Microservices Recipes

Technology Overview

INOQ

practical-microservices.com

[practical-microservices.com/
recipes.html](https://practical-microservices.com/recipes.html)



International Software Architecture
Qualification Board

<https://www.isaqb.org/>

FREE



<https://leanpub.com/service-mesh-primer/>

Wikipedia

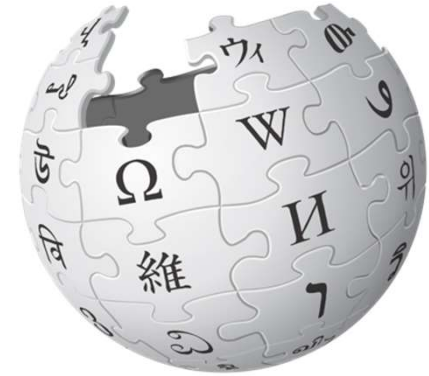
Software architecture =
high level structures

Structure =

Software elements +

Relations among them +

Properties of elements and relations



Martin Fowler

Software architecture =
Important
and hard to change decisions



Photo: Webysther Nunes

Architects do architecture
& technology!

Organization? 🙄

Conway's Law

Architecture

copies

communication structures

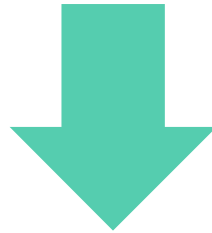
of the organization

Organization



Architecture

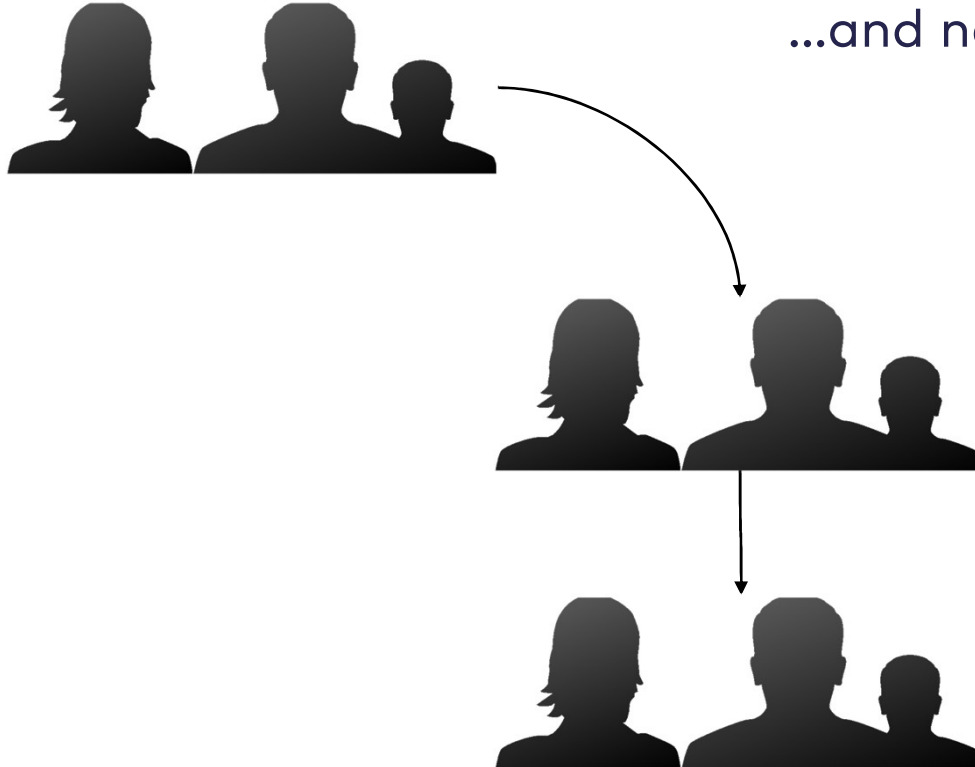
Organization



Architecture

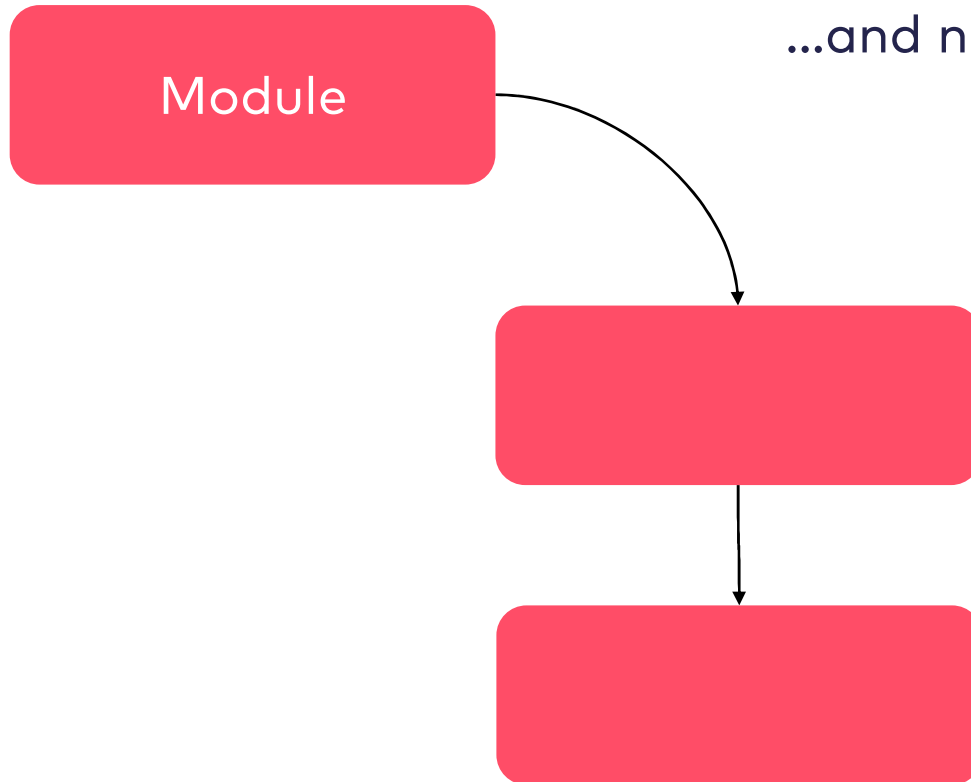
Organization

Teams will build modules
...and negotiate interfaces



Architecture

Teams will build modules
...and negotiate interfaces



Organization



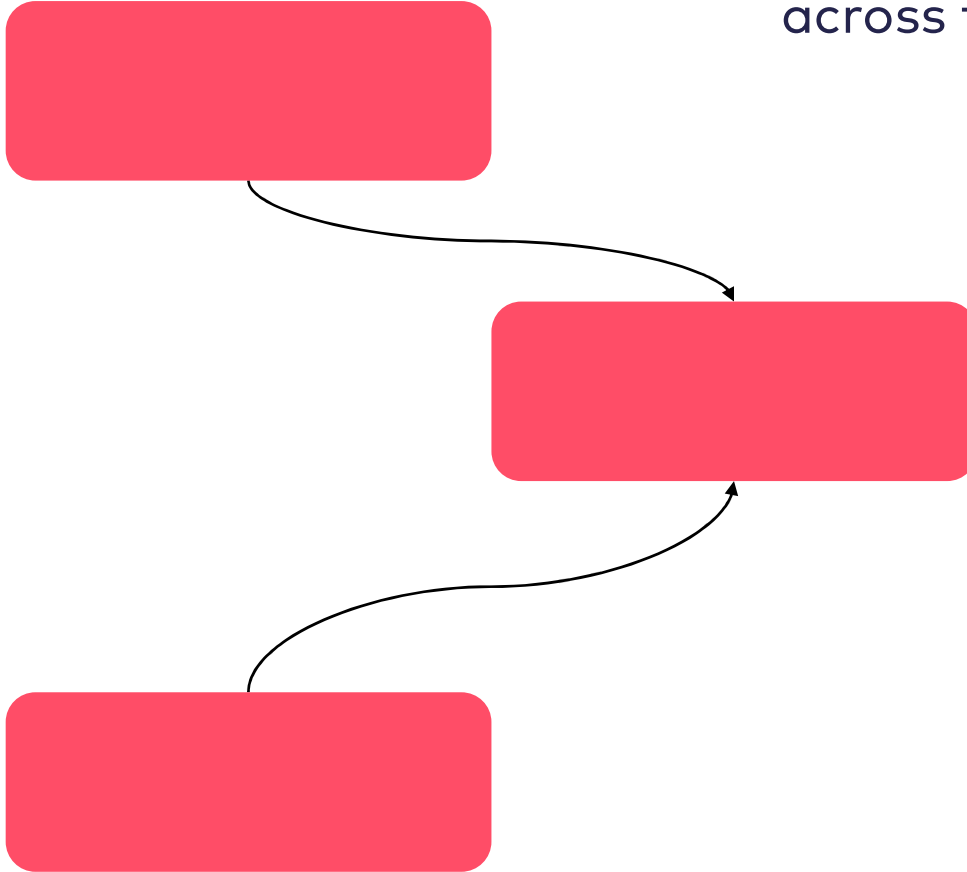
Architecture

Inverse Conway Maneuver

Microservices

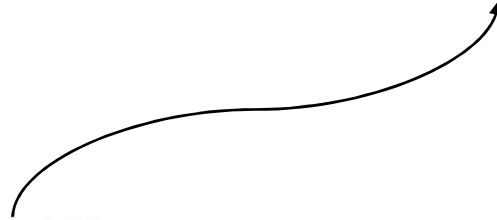
Architecture

Distribute modules
across teams

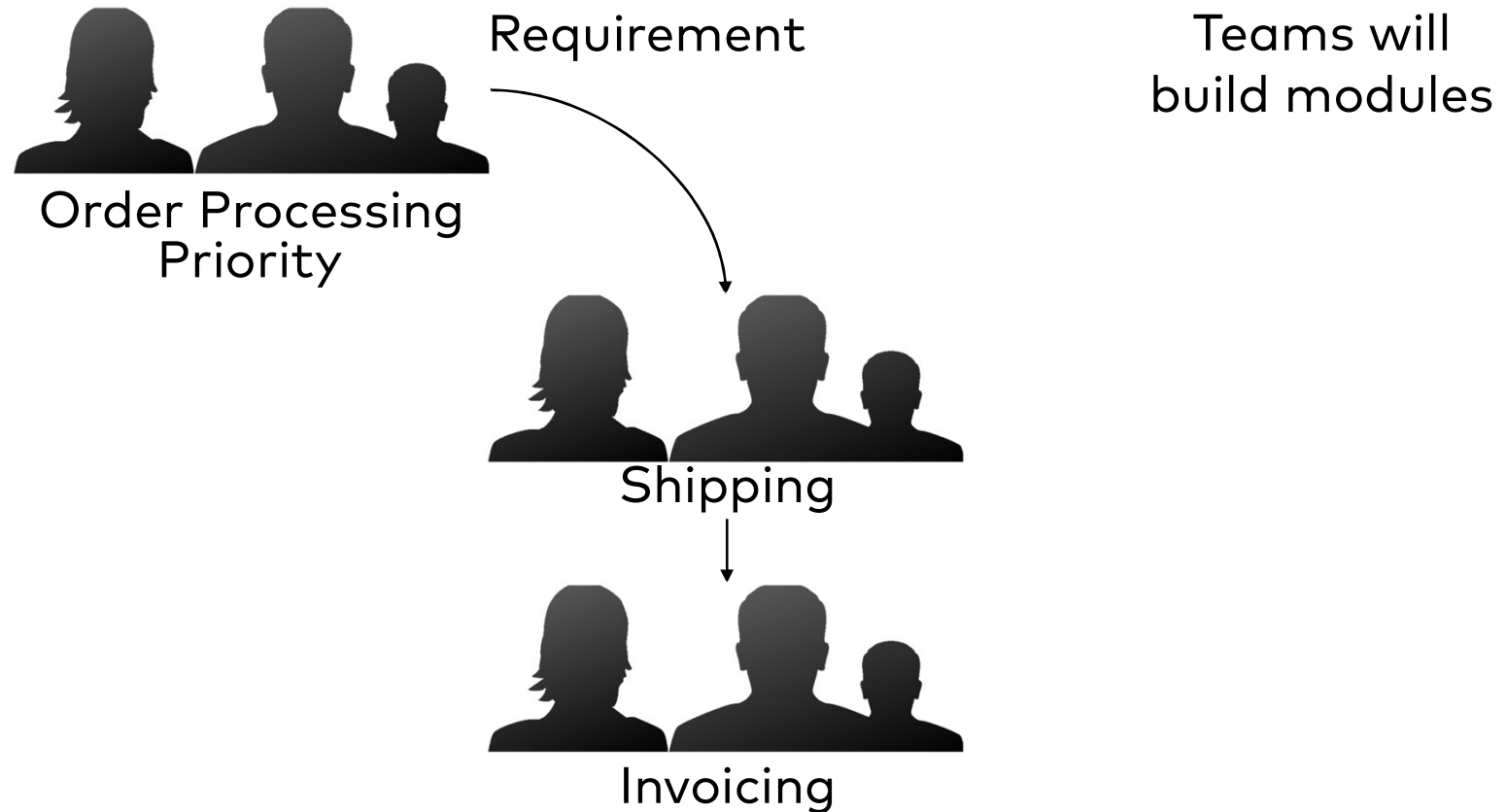


Organization

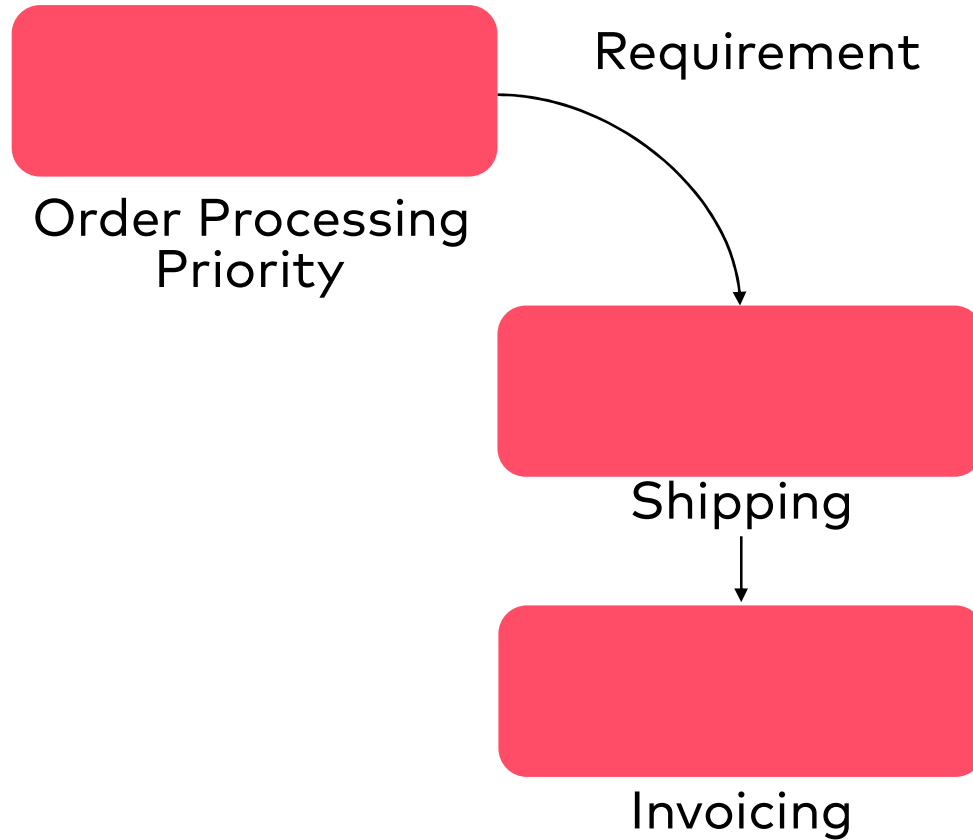
Distribute modules
across teams



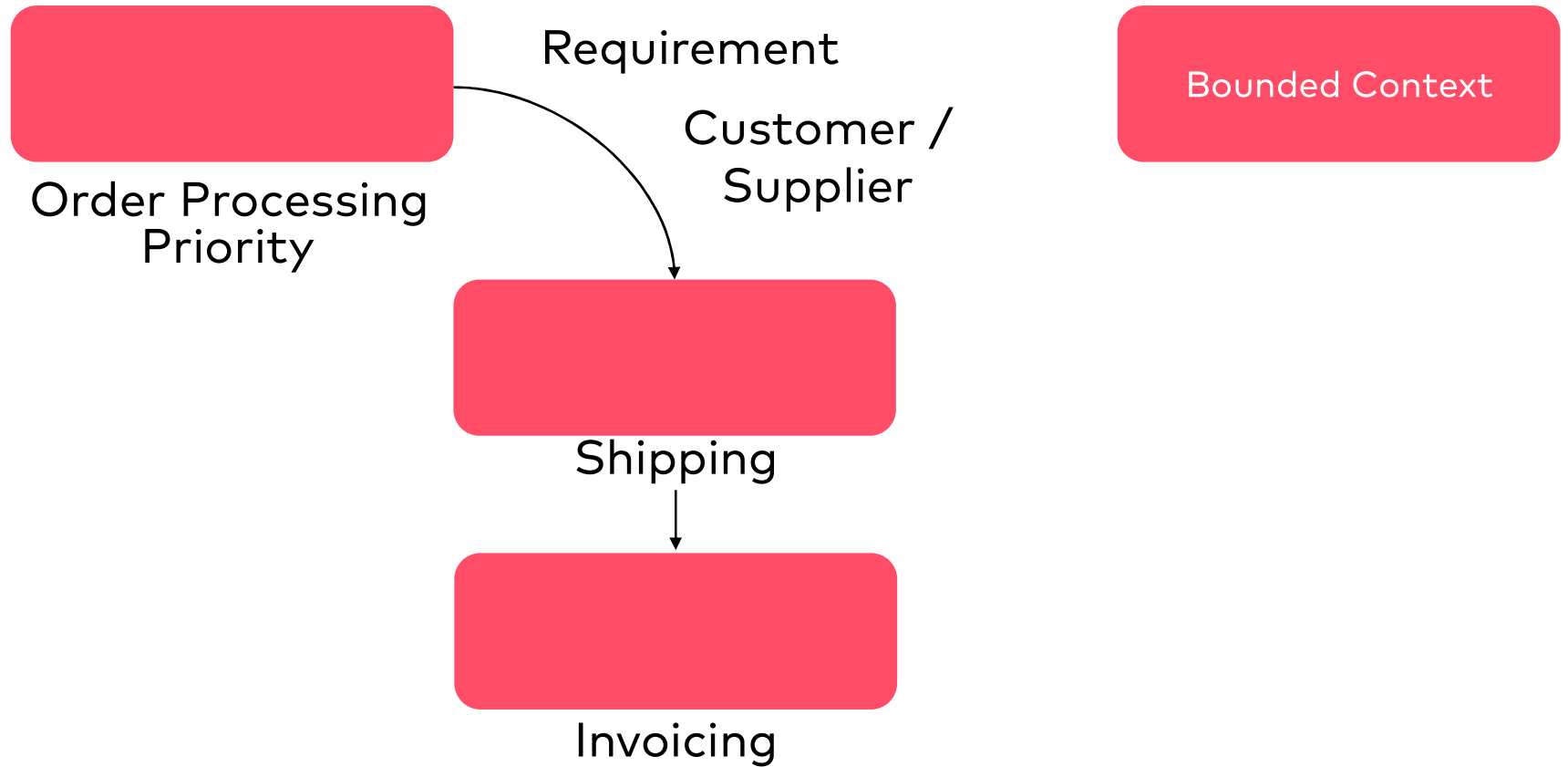
Organization: A Tool for Architecture



Organization: A Tool for Architecture



Organization: Domain-driven Design



Organization = Architecture

Conway's Law (1967)

Inverse Conway Maneuver
(Microservices 2012)

Domain-driven Design (2004)

Organization must be defined.
=coarse-grained architecture.

Managers work on organization.

Organization = architecture

Architects = managers?

Conway's Law

Architecture

copies

communication structures

of the organization

Actually, architecture = communication

Not org chart

Communication: A Tool for Architecture

Make sure the right people talk to each other

e.g. about interfaces

...or other decisions

=architecture



Start or support CoPs (Community of Practice)

e.g. for frontend, architecture etc

No need to be a manager for that....

Communication: A Tool for Architecture

Not necessarily content

...but communication

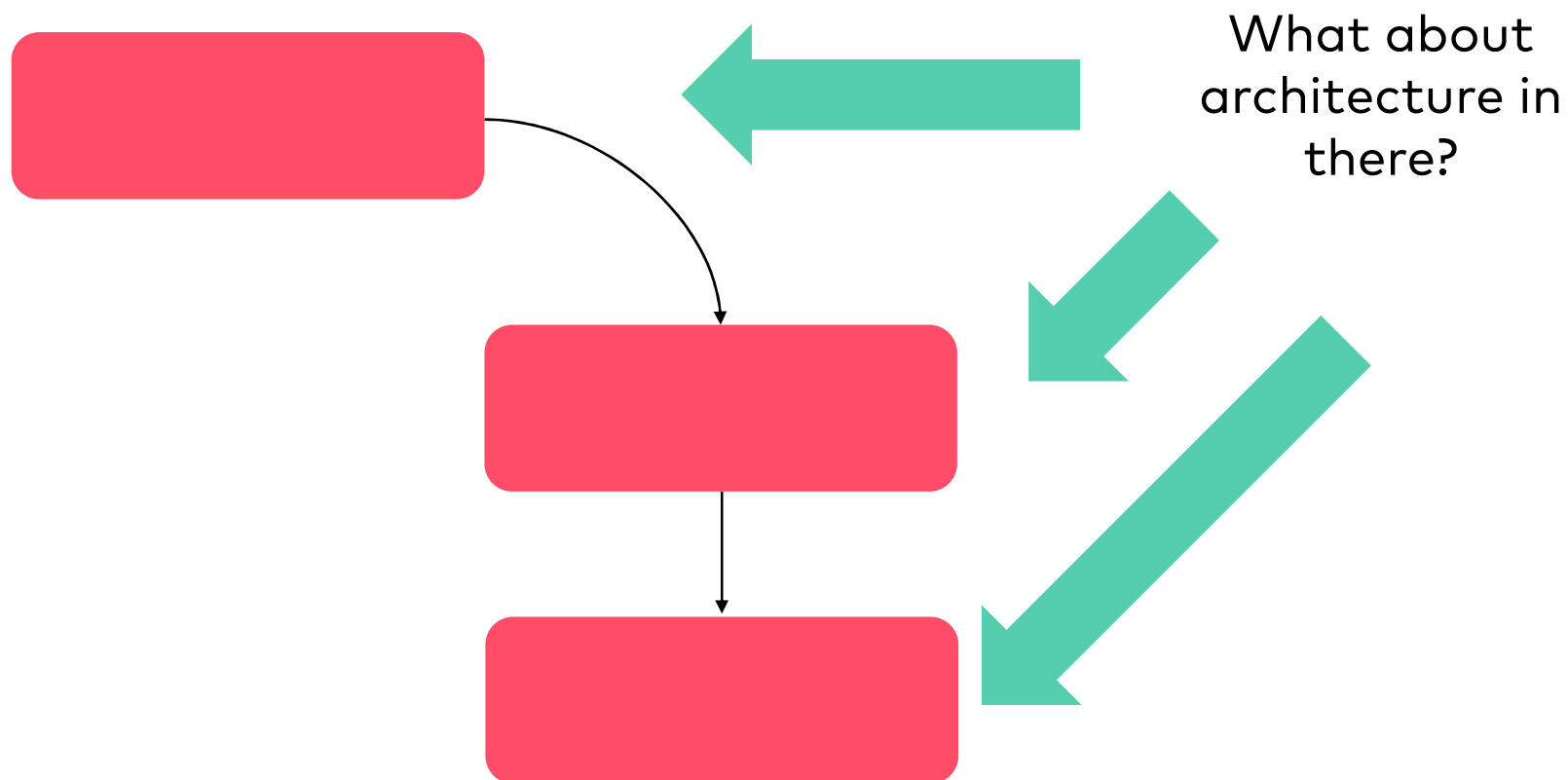
Content will follow



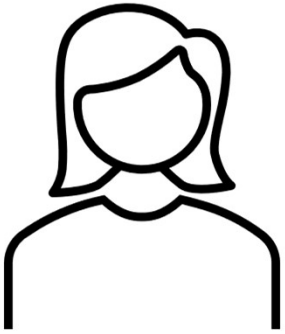
Architecture = coarse-grained decisions

Communication helps to understand details

...and communicate architecture decisions and challenges

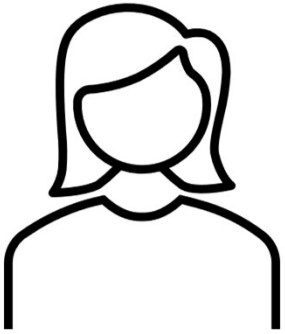


Traditional Architecture



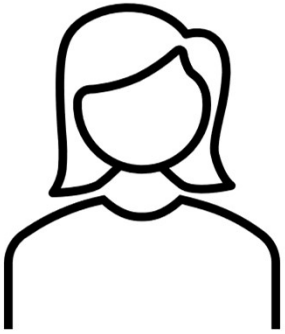
- Let the architect decide everything!
- Architect will be overloaded
- Architect cannot possibly know all details.

Architect

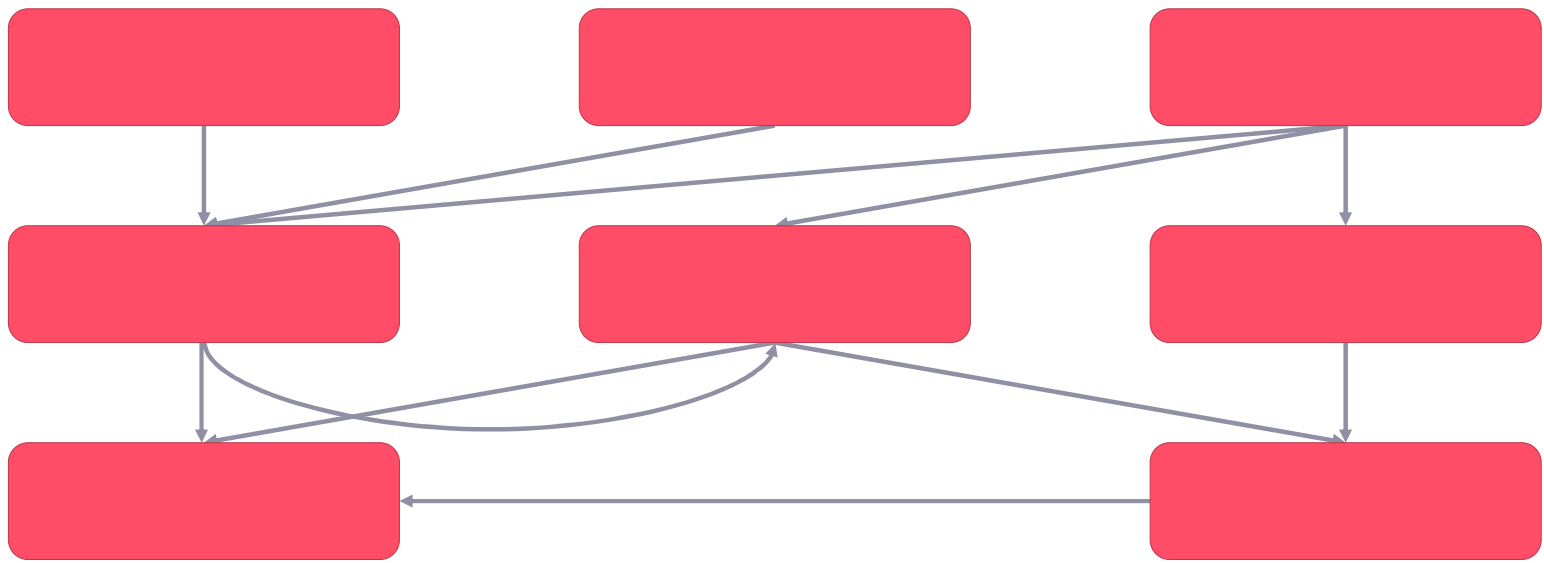


- So I want to introduce
microservices
domain-driven design
...
- How?

Architect



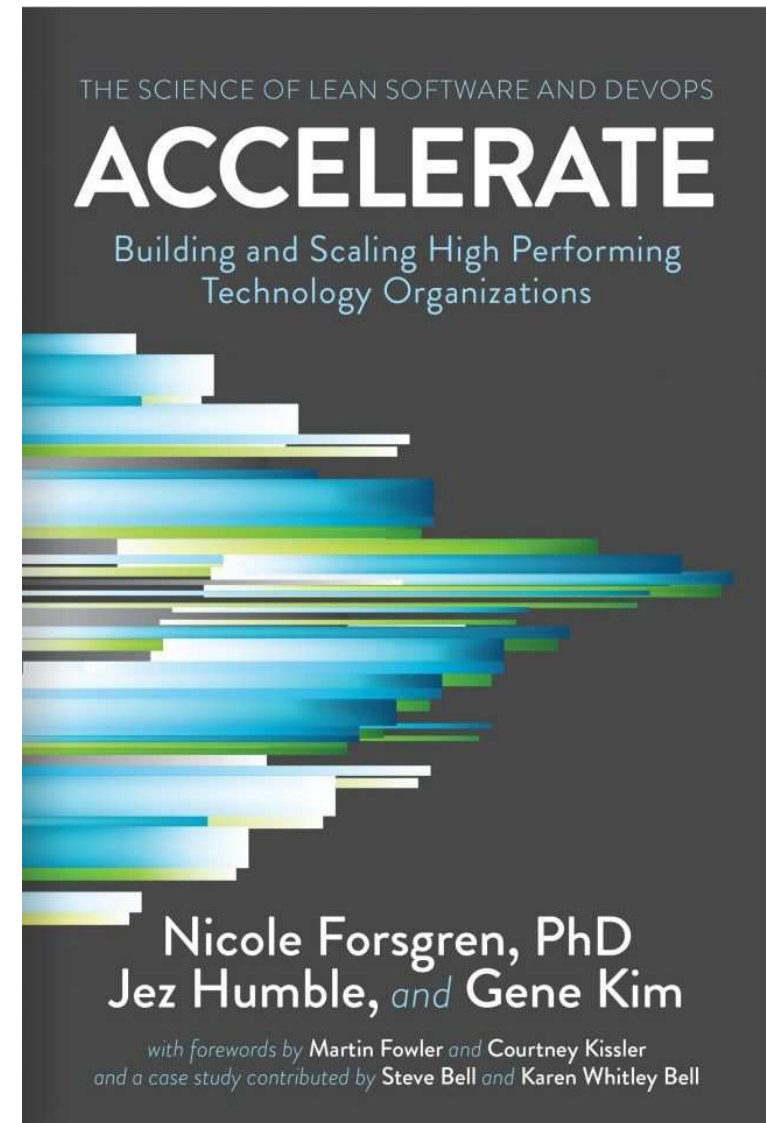
- I can just do it!
- Here is my kewl architecture!



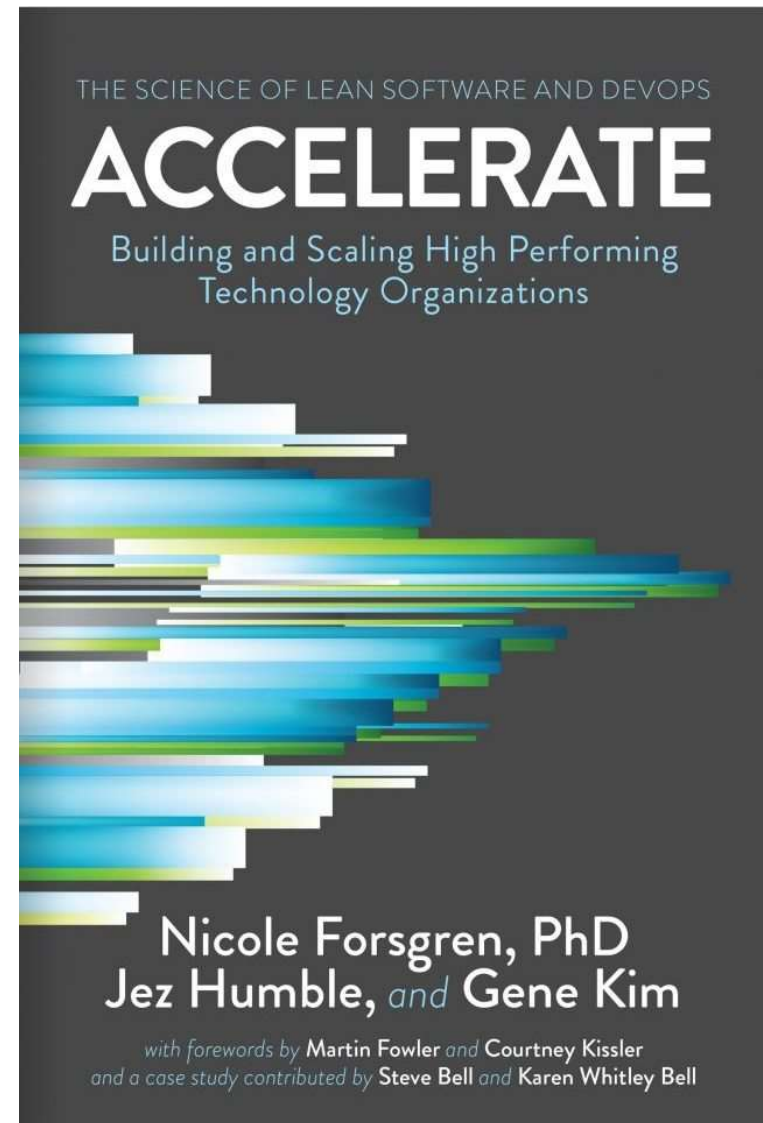
Discussion ... often focus on tools & technologies.

- Should the organization adopt **microservices**?
- Serverless?
- Kubernetes?
- Mesos?

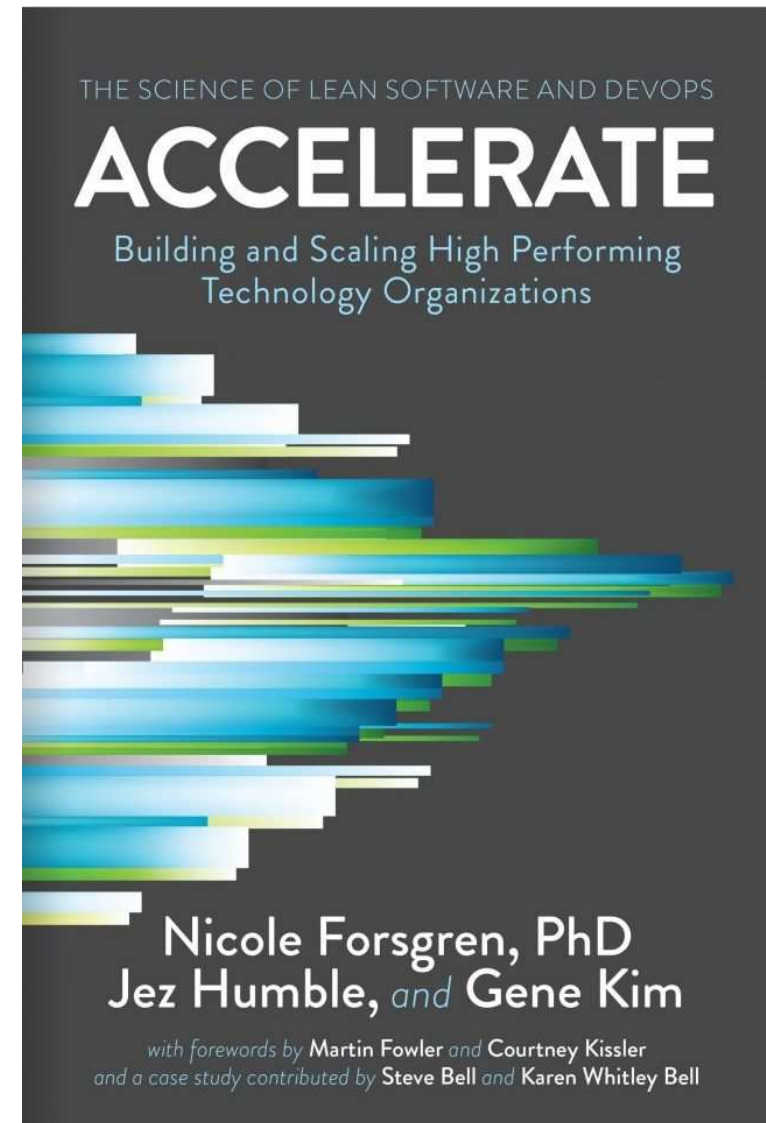
Our research shows these are the **wrong questions** to focus on.



... **tools** [are] ... **irrelevant** ...
if ... people hate them ...
or ... they don't achieve the
outcomes
and [don't] enable the
behaviors we care about.



What is important
is **enabling teams**
to make changes to their
products or services
**without depending on other
teams or systems.**





Must enable others

Team = experts

Knowledge about details of the
system

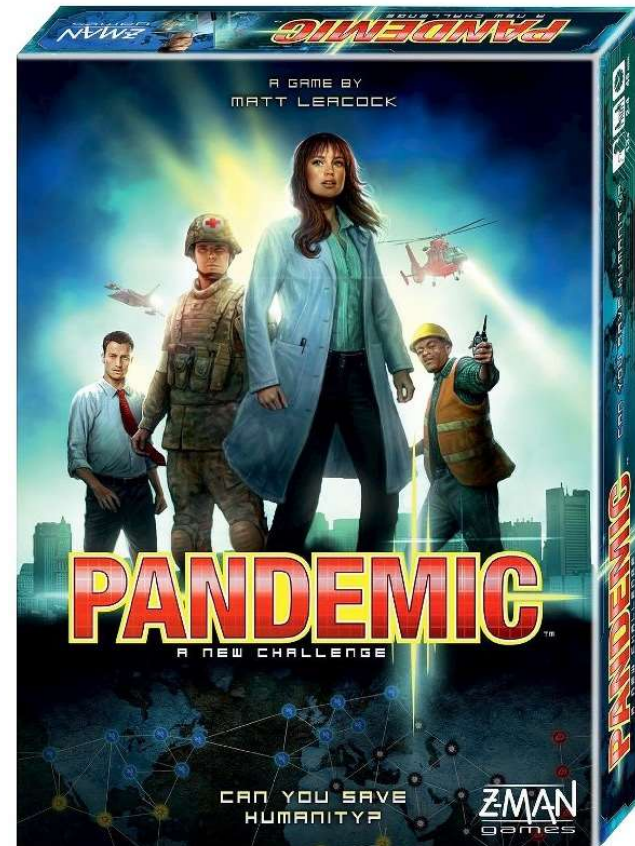
Knowledge about technologies

Software Architecture = Collaborative Game

All lose or win together

Everyone has a specific role

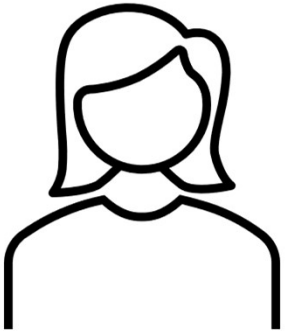
Communication is essential



**Architects shouldn't do
architecture!**

**Start: More than
Trainings**

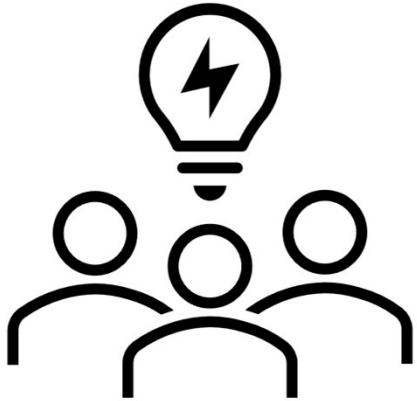
Architect



- Spread knowledge about
microservices
domain-driven design

...

- Make them use the knowledge!



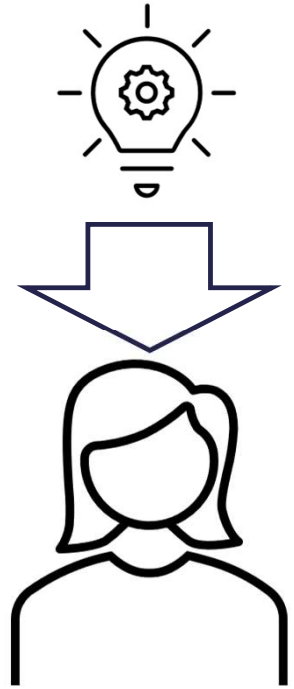
Invite diverse roles

Architecture: many stakeholders

Dev, Ops, QA, Management

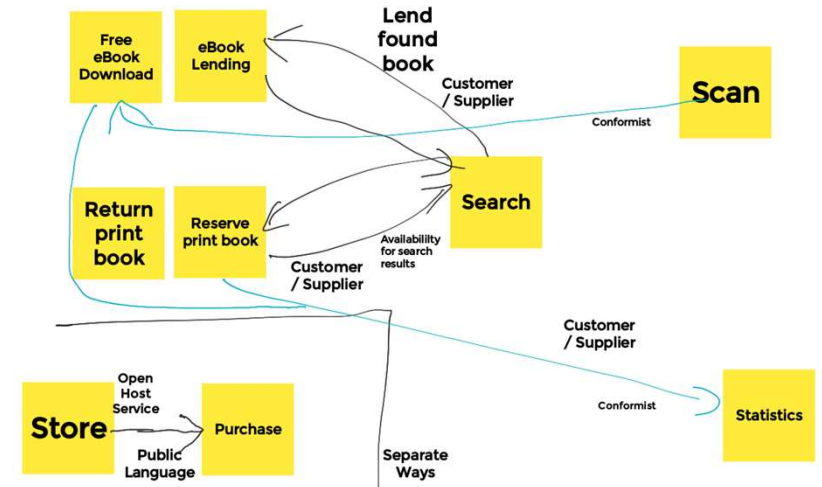
Traditional Training

- Provide knowledge
- Knowledge alone is not enough



Kata / Dojo

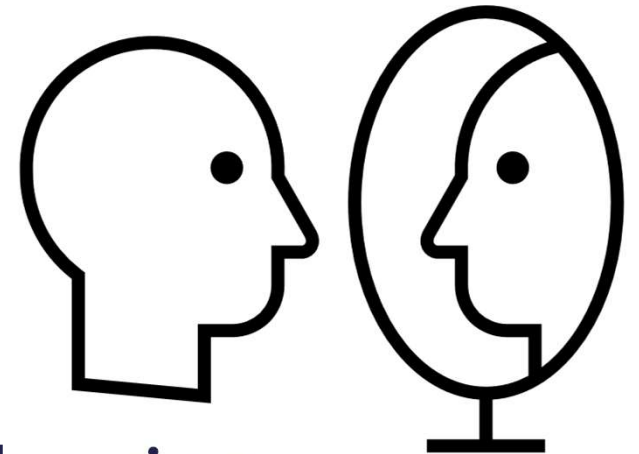
- 5 slides for 5h
- Not a lot of content
- Rest: doing labs



- Afterwards: Attendees have "done" it ...more likely to do it in the future, too!

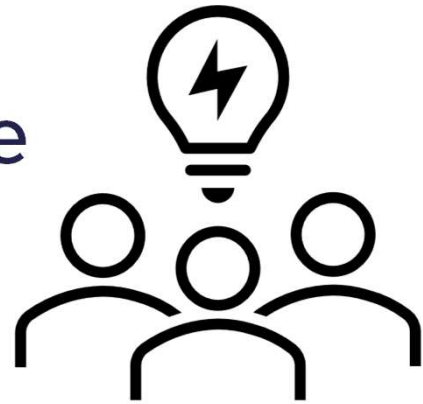
Reflection

- End of dojo / kata
- 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 Dojo / Kata 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

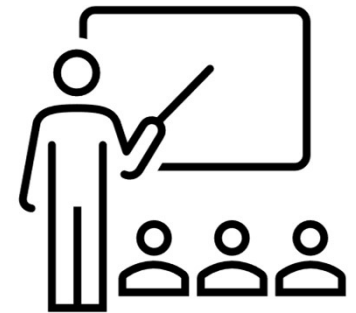


Support

Support

- Dojo / kata provides basic knowledge
...and people will try the new concepts

- They will run into challenges
- Provide consulting / support

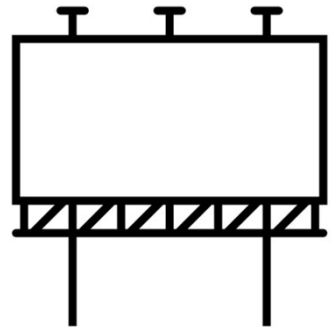


- Why should a single training be enough?

Workshop

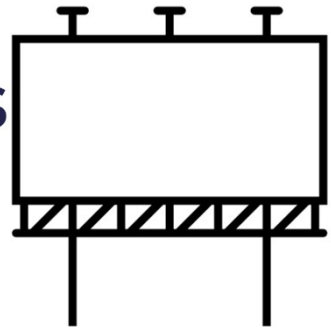
Workshop

- Someone has a challenge
e.g. how to design a part of the system
- Make it a dojo/kata!
- Write down the challenge
- Have multiple groups work on it



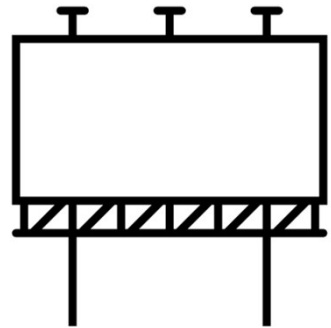
Workshop: Benefit

- Truly apply new techniques
- Spread knowledge about new techniques
- Spread knowledge about challenges
...and decisions
- Strengthen collaboration



Workshop: Challenges

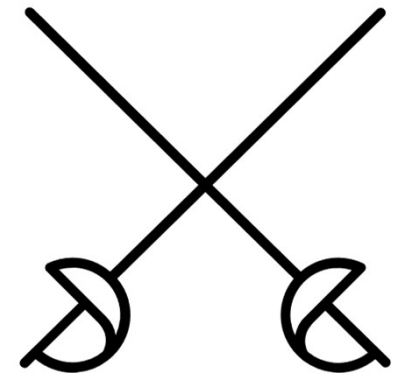
- Effort to prepare
- Instead: Open Space?
- More ad hoc workshops?



Culture Issues

Culture Issues

- Collaboration IT
domain experts
Management
....



- Culture issues will surface.

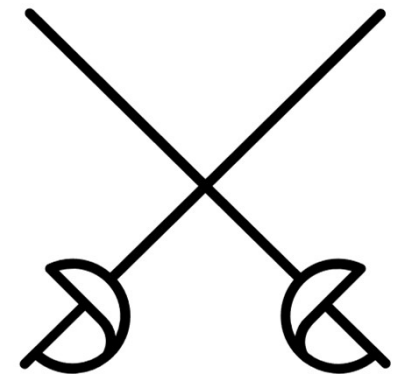
Culture Issues

- "Here is some idea!"
- "Well, just tell us how you want us to work!"

Buy-in?

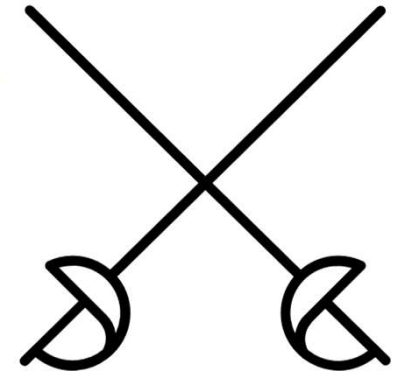
Improve the idea?

We "just" have to collaborate
...and win together



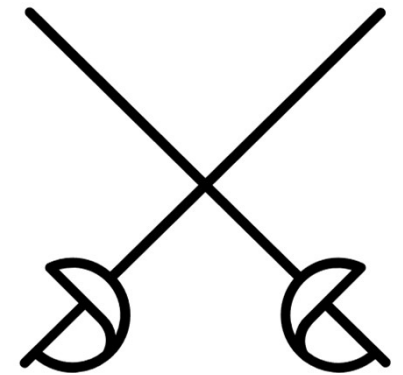
Culture Issues

- Architecture must support goals!
- "Why share our secret business strategy with architects?"
- Architecture = organization
- Management buy-in!



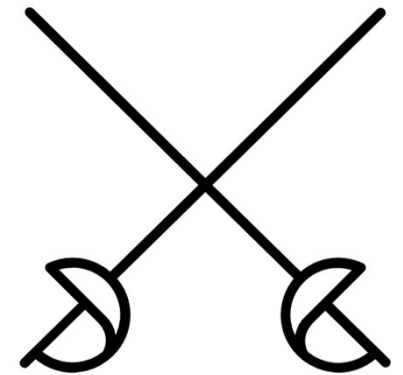
Culture Issues

- Goal: value to customer
- Where are actual users?
...or at least stakeholder outside IT?
- Direct user-feedback: strong influence on project success



Culture Issues

- Social organizations are not deterministic.
- You might try & fail.
- The same thing after a while might work.



Conclusion

Conclusion

Architect shouldn't do architecture

Defining teams =
Defining the architecture

Conclusion

Establishing communication =
Supporting architecture

Training + consulting =
Fine-grained architecture



Eberhard Wolff 

@ewolff

"We are uncovering better ways of [#developing](#)



Eberhard Wolff ✓

@ewolff

"We are uncovering better ways of [#developing](#)
[#software](#) by doing it and helping others do it."



Eberhard Wolff ✓

@ewolff

"We are uncovering better ways of [#developing](#)
[#software](#) by doing it and helping others do it."
The first sentence of the [#agile](#) manifesto.



Eberhard Wolff ✓

@ewolff

"We are uncovering better ways of [#developing](#)
[#software](#) by doing it and helping others do it."
The first sentence of the [#agile](#) manifesto.
Should be something like a "[#categorical](#)
imperative" for software engineering.



Eberhard Wolff ✓

@ewolff

"We are uncovering better ways of [#developing](#)
[#software](#) by doing it and helping others do it."
The first sentence of the [#agile](#) manifesto.
Should be something like a "[#categorical](#)
imperative" for software engineering.

11:13 AM · May 1, 2020 · [TweetDeck](#)

Send email to GOTOpia2021@ewolff.com

Slides

- + Service Mesh Primer (EN)
- + Microservices Primer DE / EN
- + Microservices Recipes DE / EN
- + DDD Reference DE
- + 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,
wrong address handled manually

