How Do We Want to Develop Software?

Eberhard Wolff

Head of Architecture

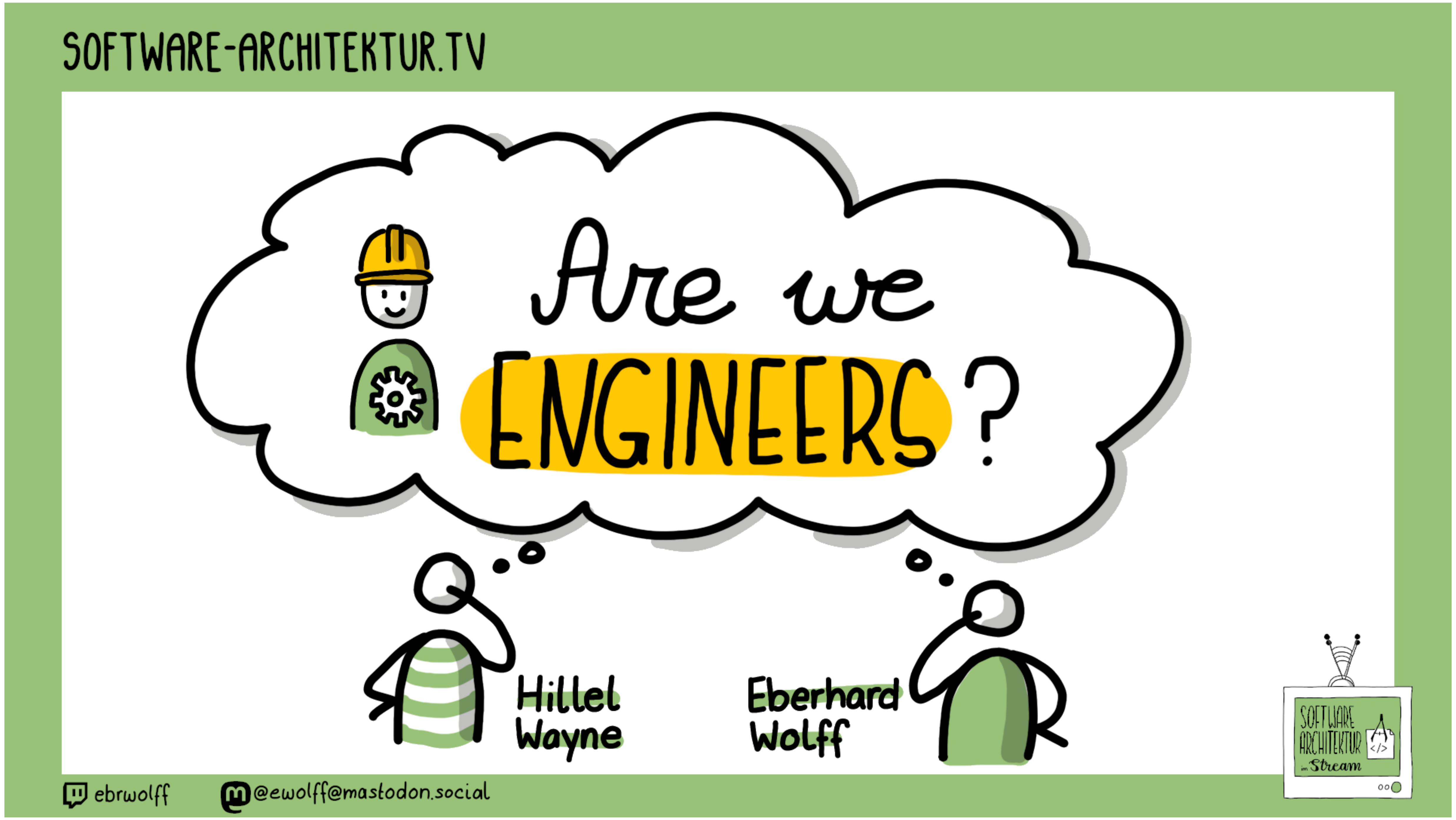
https://swaglab.rocks/
https://ewolff.com/

SWAGLab

How Do We Want to Develop Software?

- How would we like to work?
- When are we comfortable?
- Some food for thought

Our Community: Unlike Any Engineering Community!



https://software-architektur.tv/2024/03/27/folge209.html

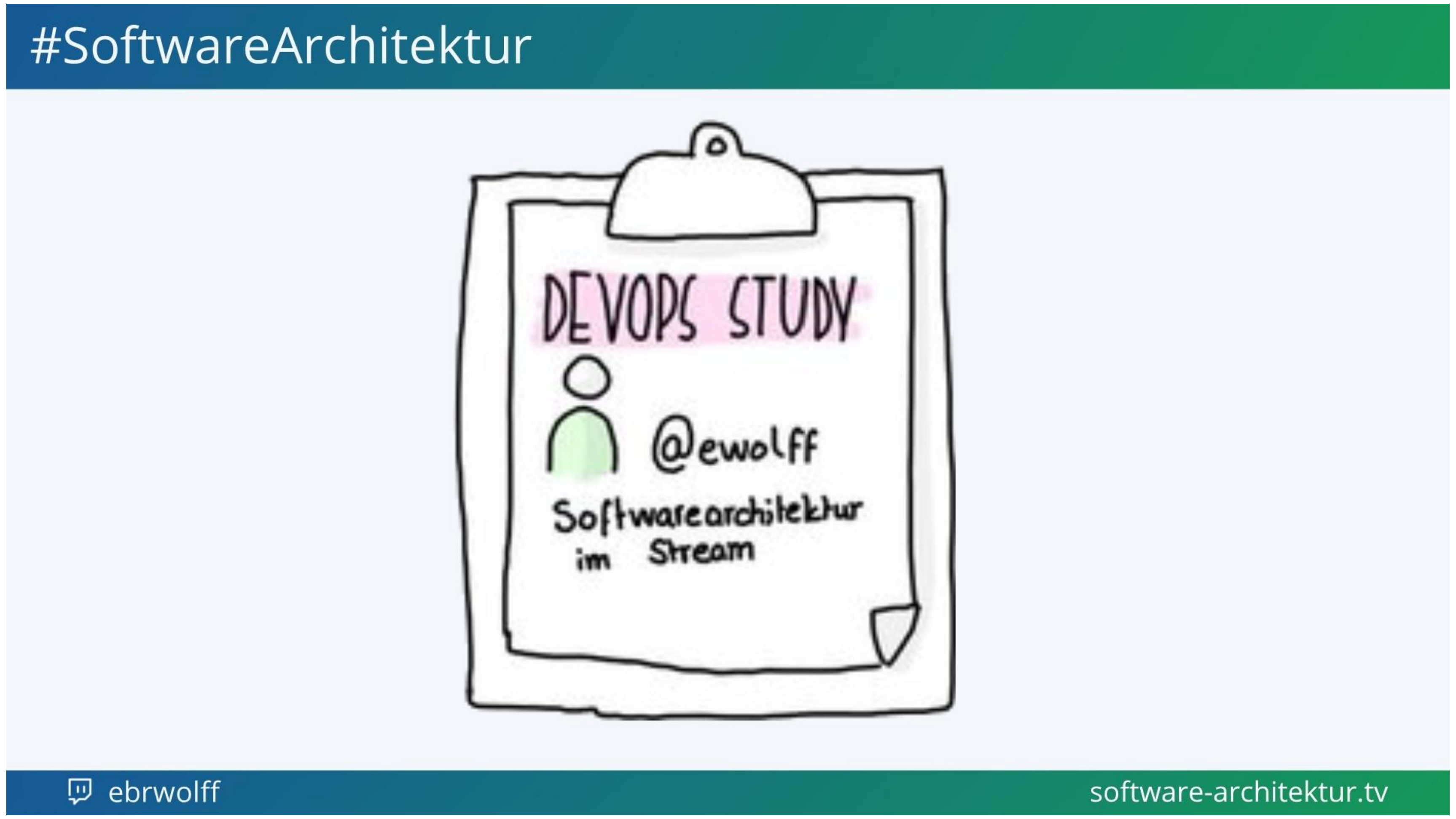
SWAGLab

Would you rather work in an environment with Continuous Delivery?

Do you think Continuous Delivery reduces burnout?

Do you think Continuous Delivery means more time for new work?

Continuously delivering new features sound like a lot of pressure?



#SoftwareArchitektur

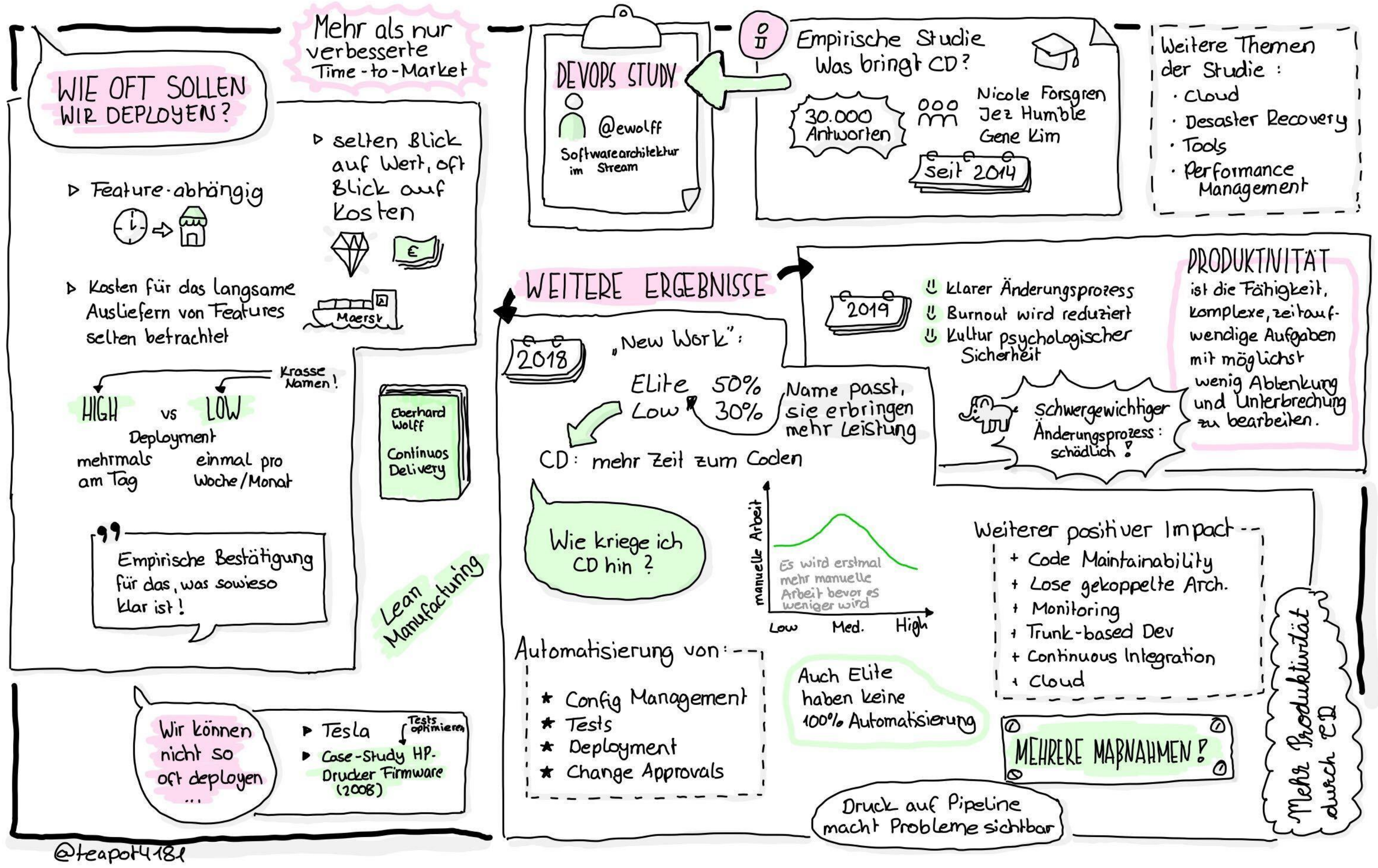
DEVOPS STUDY

@ewolff

Softwarearchitektur im Stream

ebrwolff

software-architektur.tv


WIE OFT SOLLEN WIR DEPLOYEN?
Mehr als nur verbesserte Time-to-Market
▷ Feature-abhängig
▷ selten Blick auf Wert, oft Blick auf Kosten
€
▷ Kosten für das langsame Ausliefern von Features selten betrachtet
Maersk
Krasse Namen!
HIGH vs LOW
Deployment
mehrmals am Tag
einmal pro Woche/Monat
Eberhard Wolff
Continuos Delivery
Empirische Bestätigung für das, was sowieso klar ist!
Lean Manufacturing
Wir können nicht so oft deployen ...
▶ Tesla
Tests optimieren
▶ Case-Study HP-Drucker Firmware (2008)
@teapot418
DEVOPS STUDY
@ewolff
Softwarearchitektur im Stream
Empirische Studie
Was bringt CD?
30.000 Antworten
Nicole Forsgren
Jez Humble
Gene Kim
seit 2014
Weitere Themen der Studie:
· Cloud
· Desaster Recovery
· Tools
· Performance Management
WEITERE ERGEBNISSE
2018
„New Work":
Elite 50%
Low 30%
Name passt, sie erbringen mehr Leistung
CD: mehr Zeit zum Coden
2019
klarer Änderungsprozess
Burnout wird reduziert
Kultur psychologischer Sicherheit
PRODUKTIVITÄT
ist die Fähigkeit, komplexe, zeitaufwendige Aufgaben mit möglichst wenig Ablenkung und Unterbrechung zu bearbeiten.
schwergewichtiger Änderungsprozess: schädlich!
Wie kriege ich CD hin?
manuelle Arbeit
Es wird erstmal mehr manuelle Arbeit bevor es weniger wird
Low
Med.
High
Automatisierung von:
★ Config Management
★ Tests
★ Deployment
★ Change Approvals
Auch Elite haben keine 100% Automatisierung
Weiterer positiver Impact
+ Code Maintainability
+ Lose gekoppelte Arch.
+ Monitoring
+ Trunk-based Dev
+ Continuous Integration
+ Cloud
MEHRERE MAßNAHMEN!
Druck auf Pipeline macht Probleme sichtbar
Mehr Produktivität durch CD

https://software-architektur.tv/2020/08/14/folge012.html

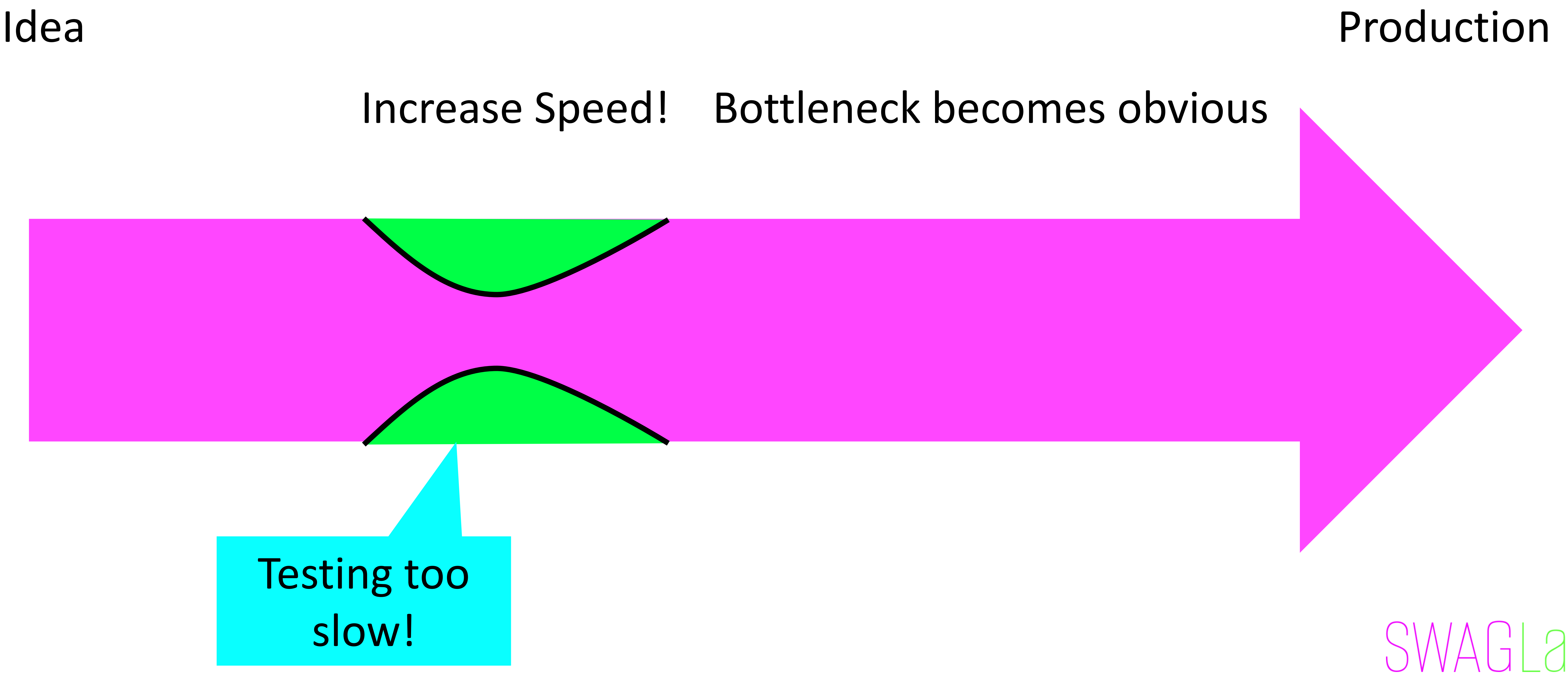
SWAGLab

DORA: 4 Key Metrics

- Change lead time
- Deployment frequency
- Change fail percentage
- Failed deployment recovery time

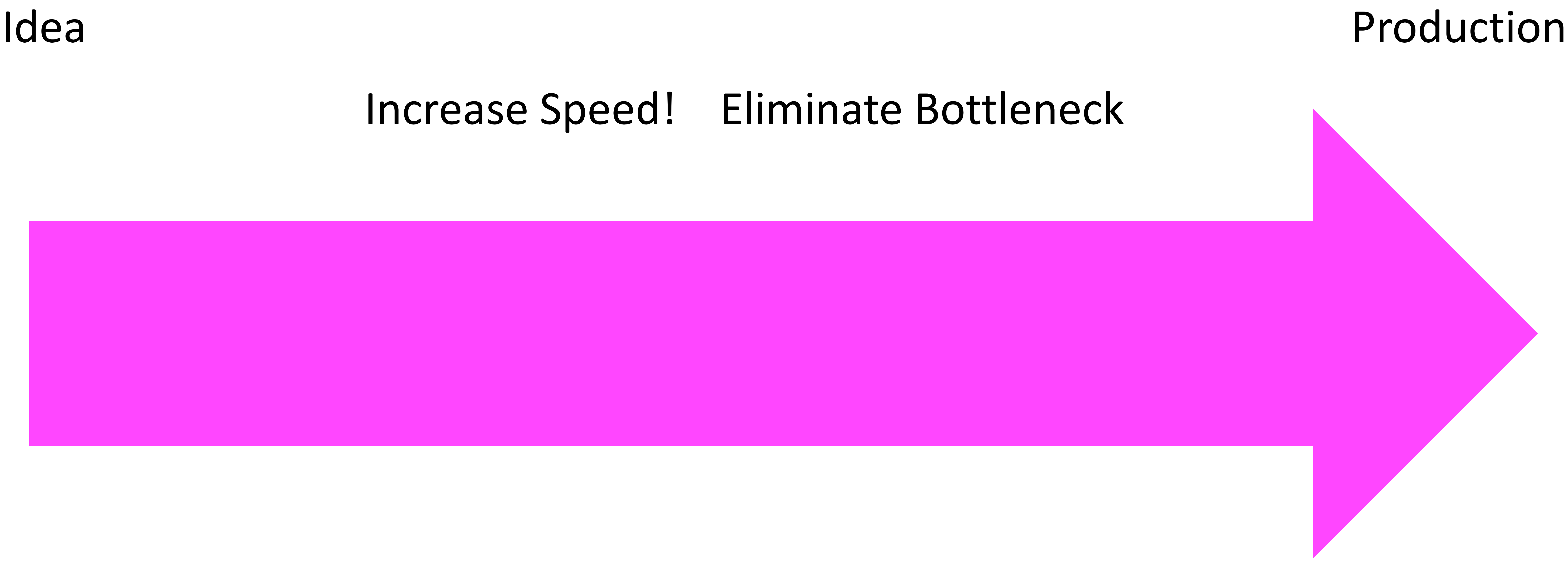
https://dora.dev/

DORA: Lean Optimization



SWAGLab

DORA: Lean Optimization



Idea

Production

Increase Speed!

Eliminate Bottleneck

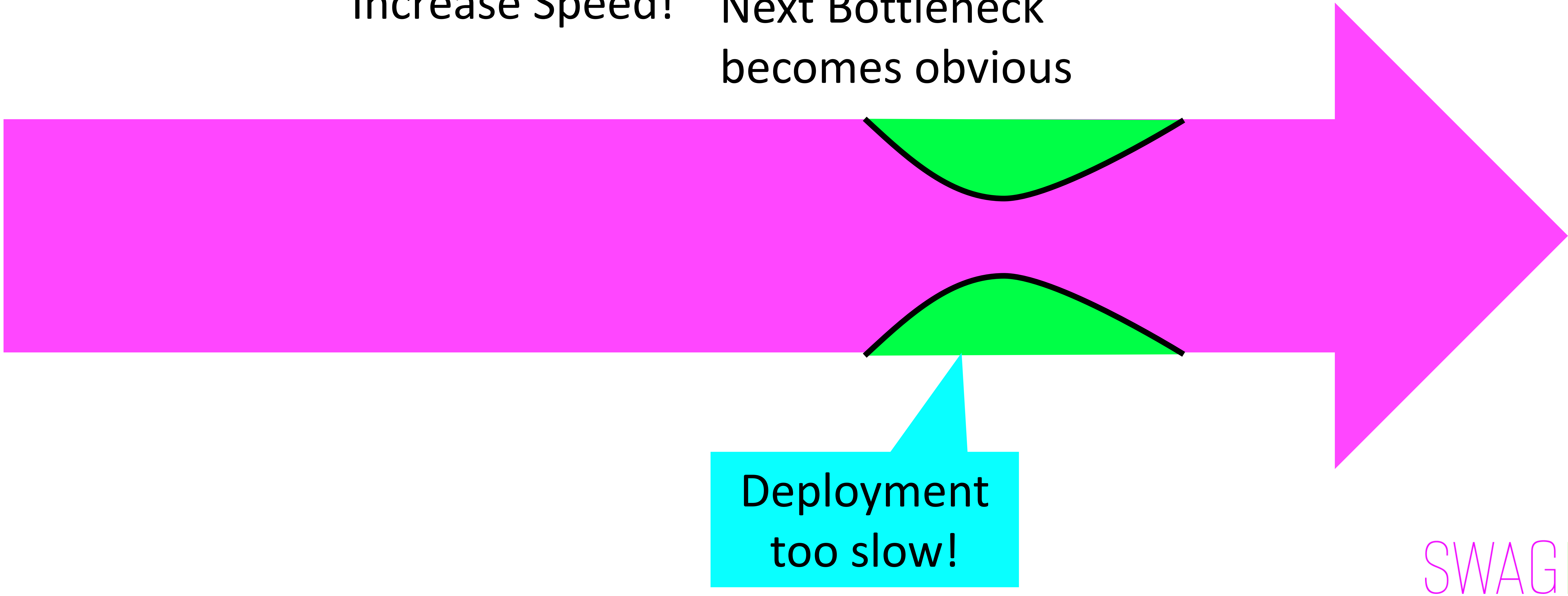
DORA: Lean Optimization

Idea

Production

Increase Speed!

Next Bottleneck becomes obvious



Deployment too slow!

DORA DevOps Study: Results

- Continuous Delivery: more time for new work.
- Continuous Delivery reduces burnout.

…because deployments probably work.

…because you need a culture of psychological safety to actually improve the process.

Trust Your Guts?

- People prefer Continuous Delivery.
- But: Usually “real” benefits are not that clear.
- Even concerning burnout
- If you prefer Continunous Delivery, shouldn’t it obviously reduce burnout?

Trust Your Guts?

- I wrote a book about Continuous Delivery

…because that is how I would like to work.

…but I failed to really say so in the book.

Maybe things we prefer are actually better?

How do you measure developer productivity?

Developer Productivity

- Why would you measure developer productivity?
- Might lead to “optimizations” i.e. more pressure.
- Might lead to lay offs.
- But can also point in the right direction.

SPACE: Areas

- SPACE: framework to measure developer productivity
- Choose a specific set of metrics to understand a specific problem

SPACE: Matrix of Metrics by Levels & Areas

	Area 1	**Area 2**	**…**	
Level 1	Metric	…		
Level 2	…			
…				

SPACE: Three Levels

Individual / one person

Team or group / people that work together

System

SPACE: Areas

- Performance (Outcome)
  Code review velocity
- Activity (Count of actions)
  Code review scores

SPACE: Areas

- Satisfaction & Well-Being:
  e.g. Developer satisfaction / retention
- Communication & Collaboration
  Code review thoughtfulness
- Efficiency & Flow
  Productivity perception

SPACE: Recommendations

- Use multiple metrics across various dimensions
- At least 3, but not too many
- At least one perceptual (survey)
- Perceptual metrics are more “precise”

i.e. hard to gamble.

Developer Productivity

- Perceptual i.e. subjective metrics are recommended.
- I.e. if subjectively something is wrong, probably something is wrong.

…and it is reliable because it is hard to gamble

Well-being is “officially” a metric for developer productivity.

What is the Most Important Skill in IT?

What is the Most Important Skill in IT?

- We asked people at a technical conference.
- Just an experiment
- We did not really think about possible answers.
- By far the most common answer: communication, soft skill …

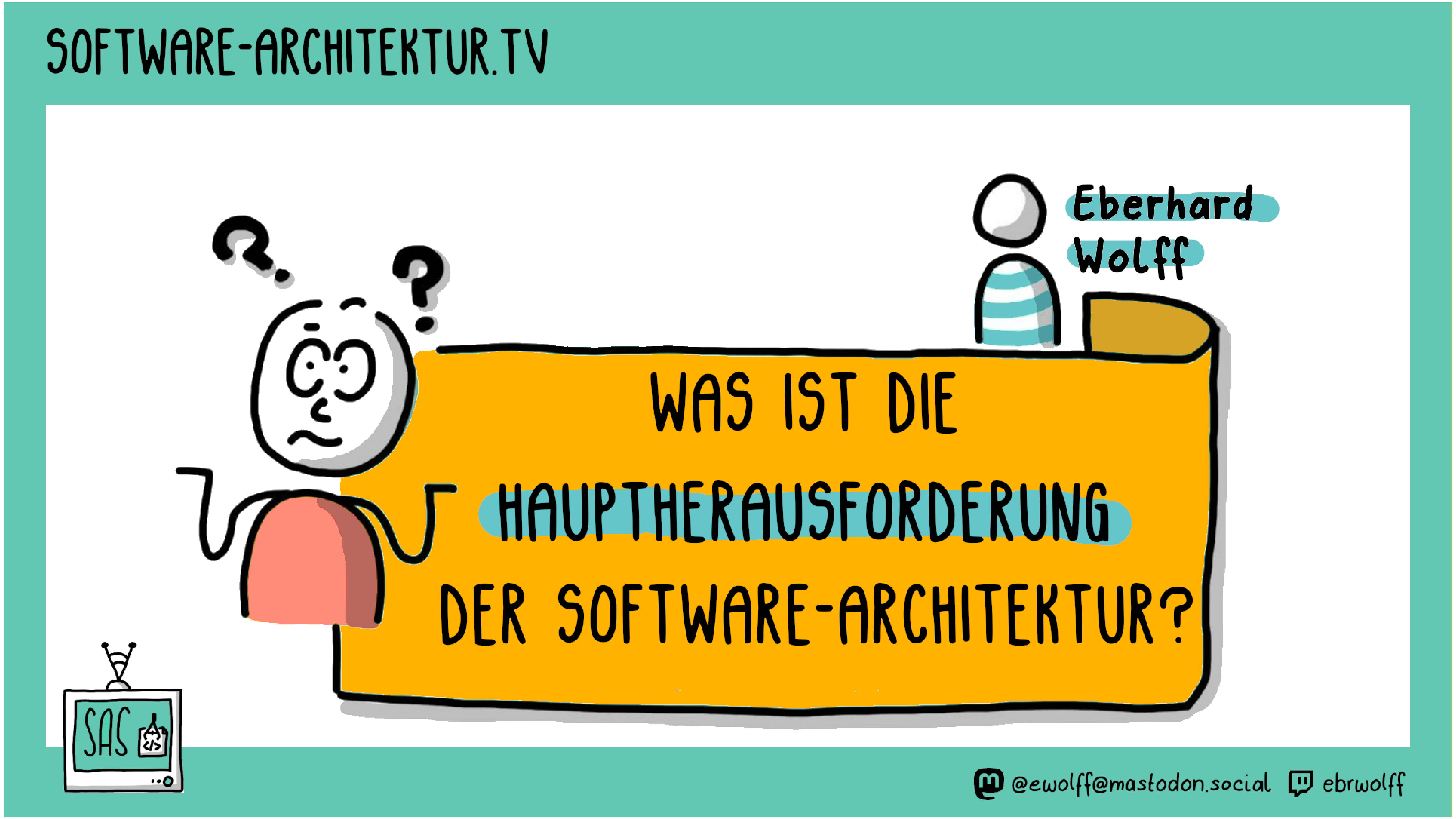
https://software-architektur.tv/2024/08/16/episode228.html

What is the Main Challenge in Software Architecture?

- Survey through social media
- What is the biggest challenge in #SoftwareArchitecture in your opinion?

Main Challenge in Software Architecture?

- Communication
- Organization and architecture
- Evolution of systems



https://software-architektur.tv/2025/06/27/folge269.html

What does that mean?

- Most important skill in IT: Communication
- Main challenge in software architecture: Rather soft skills
- Why should LLMs / AI improve productivity massively?

Development: A People Business

- Why is there such a focus on technology then?
- Will communication and collaboration work if we feel uncomfortable?
- We already discussed productivity… 🙂

Architecture

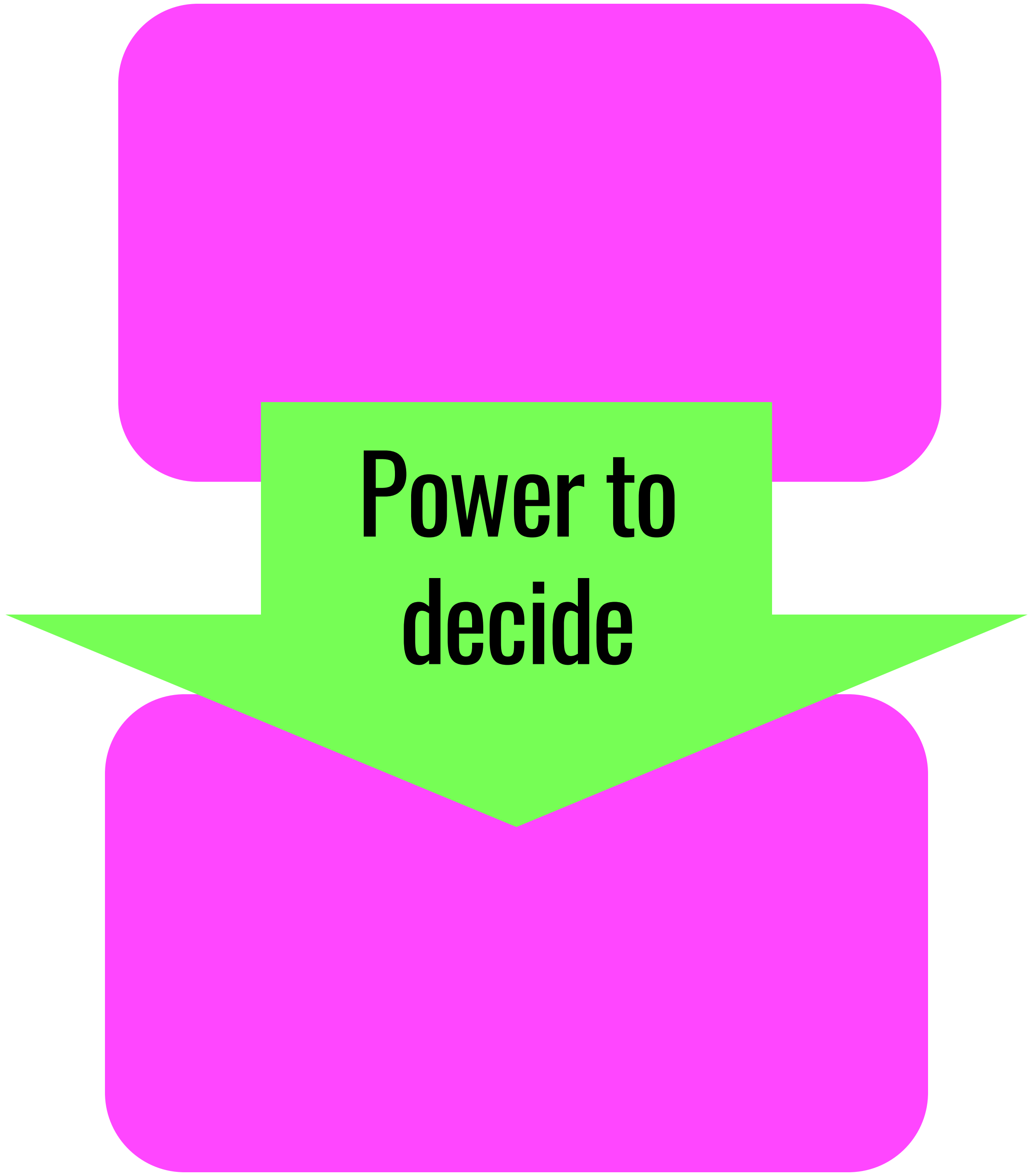
Microservices

- Make it possible to use different languages / frameworks / … in each microservices.
- Should teams be allowed to choose any technology they like?
- Every team around you uses Typescript. Does that influence your decision which technology to use yourself?

Microservices Technology Choice

- Usually: best tool for the job vs. standardization
- Standardization means sharing knowledge and make things easier.
- Are teams not aware of this?
- So why not let the teams make the trade-off?

Delegation



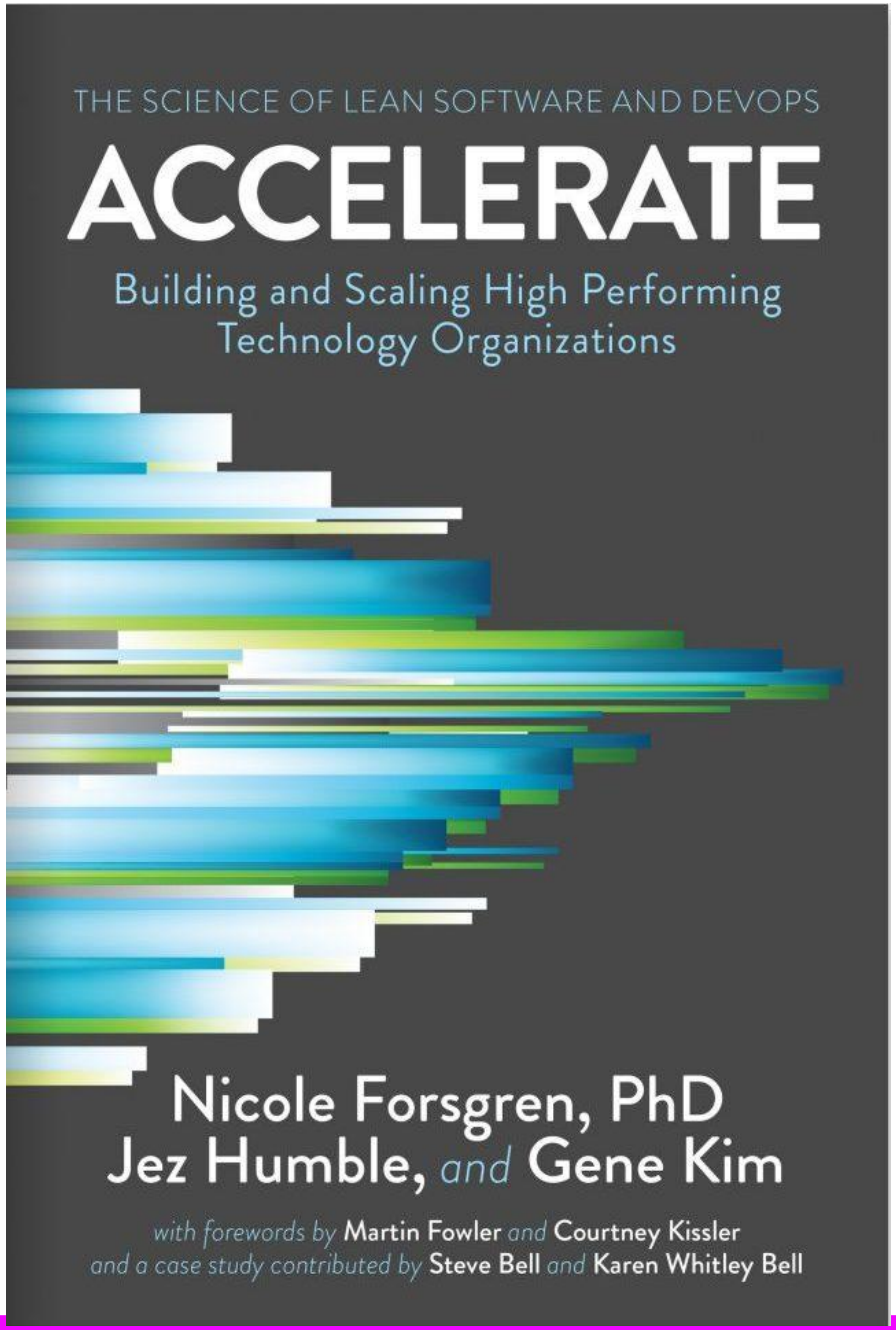
Requires trust

What is important

is enabling teams

to make changes to their products or services

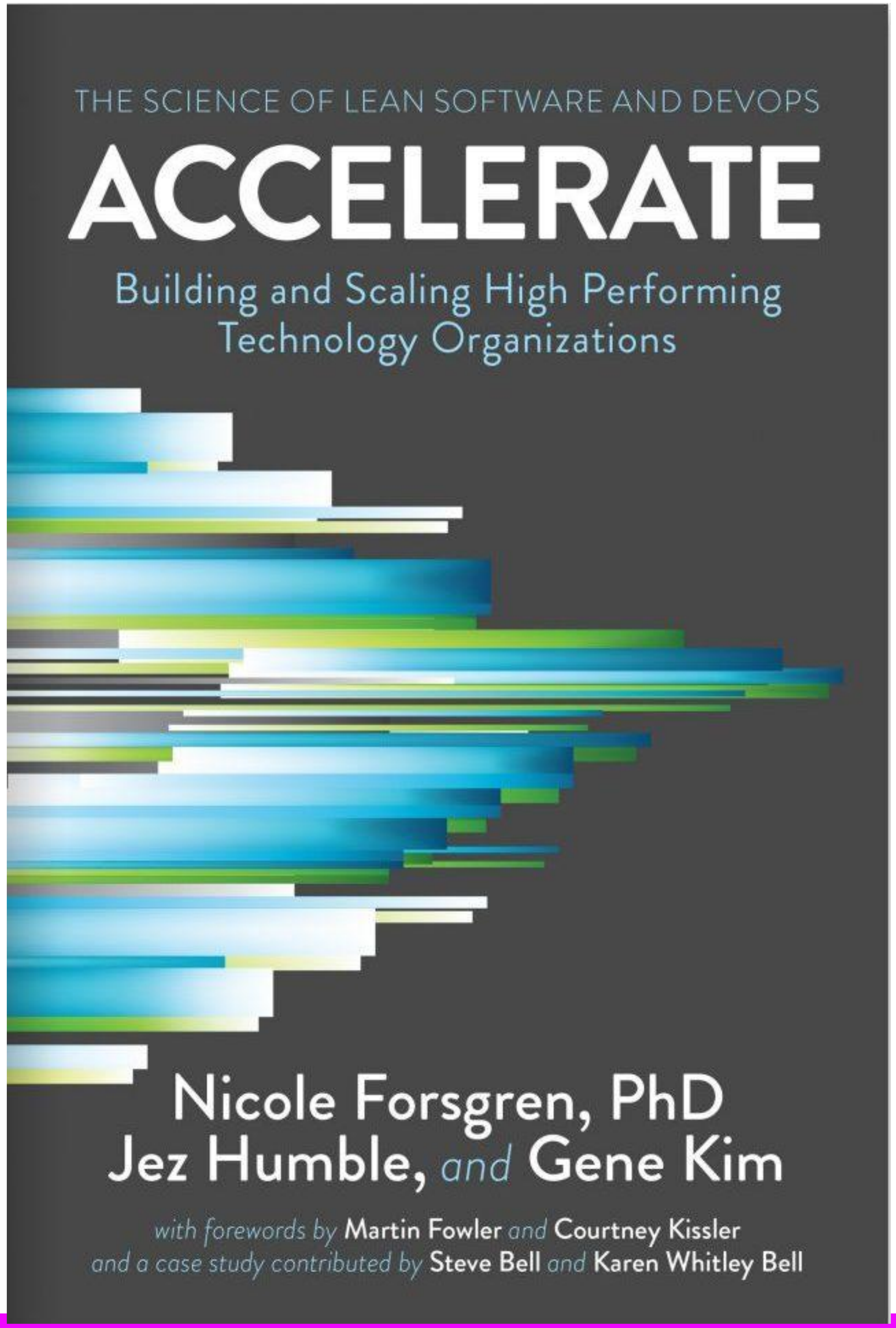
without depending on other teams or systems.


THE SCIENCE OF LEAN SOFTWARE AND DEVOPS
ACCELERATE
Building and Scaling High Performing Technology Organizations
Nicole Forsgren, PhD
Jez Humble, and Gene Kim
with forewords by Martin Fowler and Courtney Kissler
and a case study contributed by Steve Bell and Karen Whitley Bell

Discussion … often focus on tools & technologies.

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

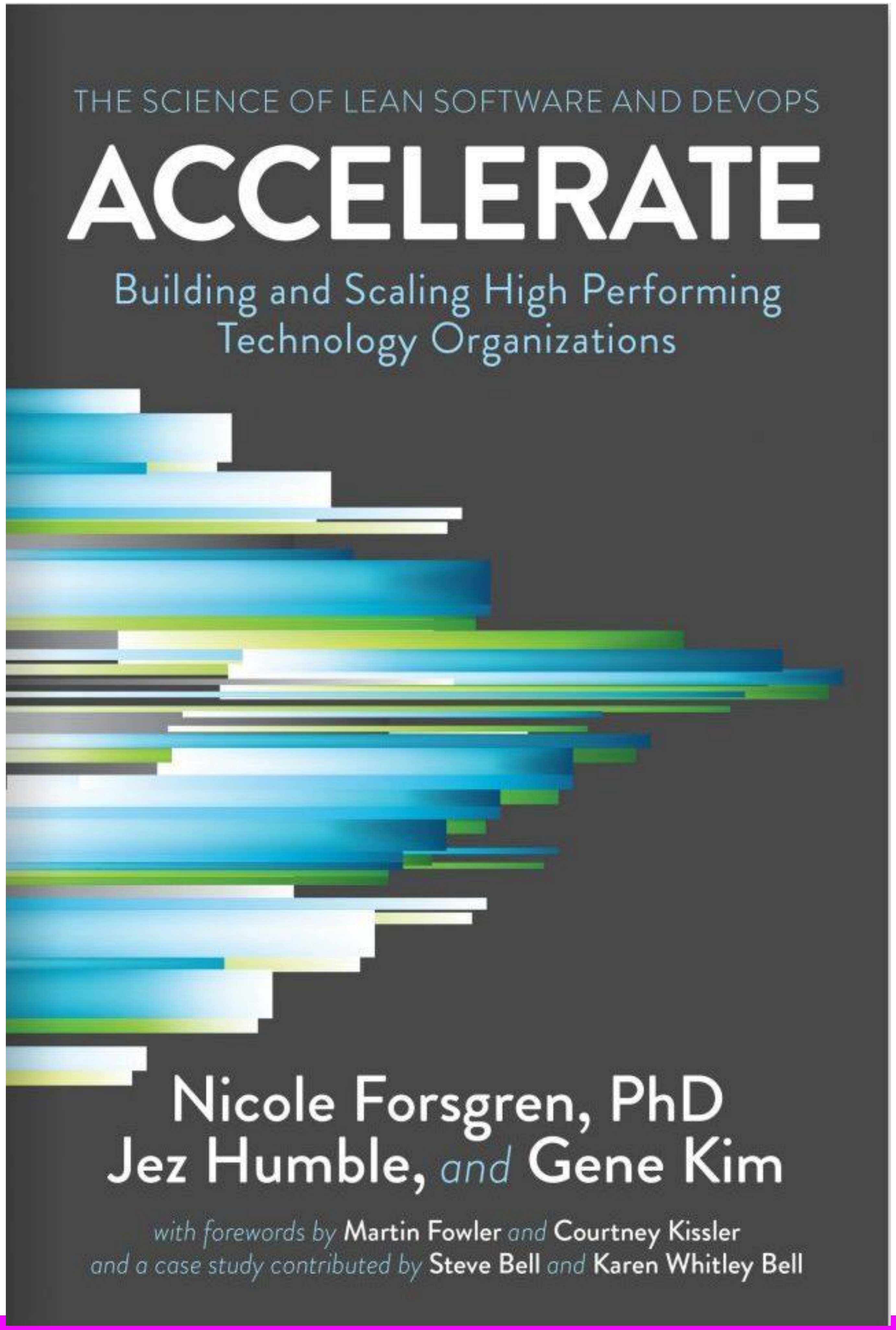
Our research shows these are the wrong questions to focus on.


THE SCIENCE OF LEAN SOFTWARE AND DEVOPS
ACCELERATE
Building and Scaling High Performing Technology Organizations
Nicole Forsgren, PhD
Jez Humble, and Gene Kim
with forewords by Martin Fowler and Courtney Kissler
and a case study contributed by Steve Bell and Karen Whitley Bell

… tools [are] … irrelevant … if … people hate them …

or … they don’t achieve the outcomes

and [don’t] enable the behaviors we care about.


THE SCIENCE OF LEAN SOFTWARE AND DEVOPS
ACCELERATE
Building and Scaling High Performing Technology Organizations
Nicole Forsgren, PhD
Jez Humble, and Gene Kim
with forewords by Martin Fowler and Courtney Kissler
and a case study contributed by Steve Bell and Karen Whitley Bell

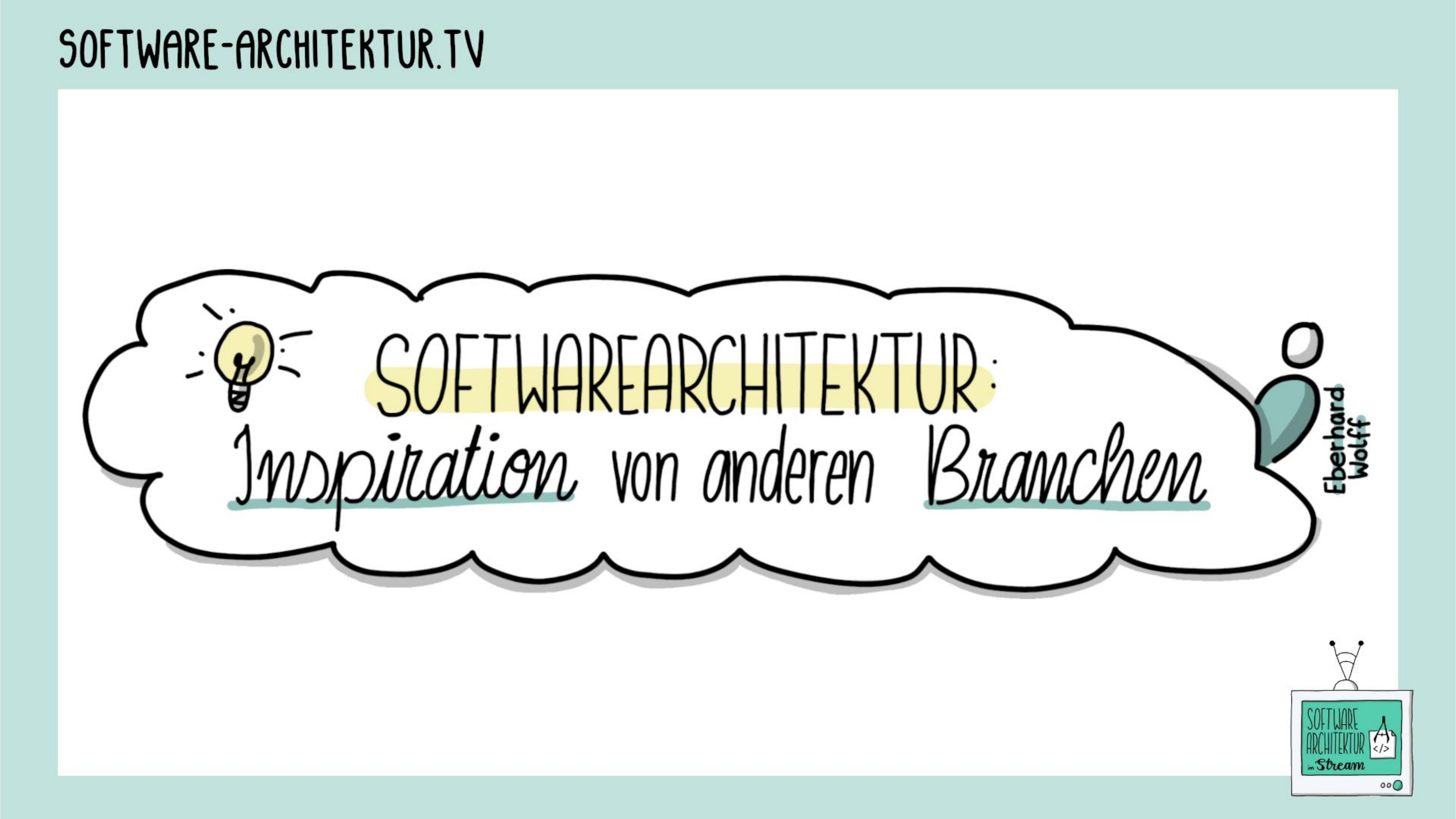
So software development should be an enjoyable party 🥳 ?

Trust

- Delegation requires trust.
- Some people you can’t trust.
- Then you will need to work differently.
- So it is important to work with the right people.
- But: You have to work with the available people.

Now What?

- Other industries systematically improve collaboration.
- Aviation: Crew Resource Management
- Military: Auftragstaktik (Mission-type tactics)
- Imagine an army commander requesting different people…


SOFTWARE-ARCHITEKTUR.TV
SOFTWAREARCHITEKTUR:
Inspiration von anderen Branchen
Eberhard Wolff
SOFTWARE ARCHITEKTUR im Stream
</>

https://software-architektur.tv/2023/10/27/folge187.html

Exploitation

- It is possible to be successful by exploiting people.
- I.e. an unpleasant place to work can be successful.
- I.e. greed will not lead to better culture.

Conclusion

Conclusion

- Follow your guts more!
- We prefer Continuous Delivery
  – it provides better productivity.
- Measuring developer productivity should include subjective metrics.

Conclusion

- It’s a people business – how can you be successful if everyone is uncomfortable?
- Architecture can enable independent decisions – but they must be allowed.
- But maybe you don’t trust someone for a good reason.
- …and exploitation is still successful.

How do we want to develop software?

But really: What environment do we want to live in?

DRINK A VIRTUAL COFFEE WITH ME!

Eberhard Wolff

Head of Architecture

https://swaglab.rocks/
https://ewolff.com/

https://calendly.com/eberhard-wolff-swaglab/

SWAGLab

Slides + more

Scan QR code
or send email to
javaforumnord2025@ewolff.com

Powered by Amazon Lambda

Email address logged for 14 days,
wrong addressed emails handled
manually