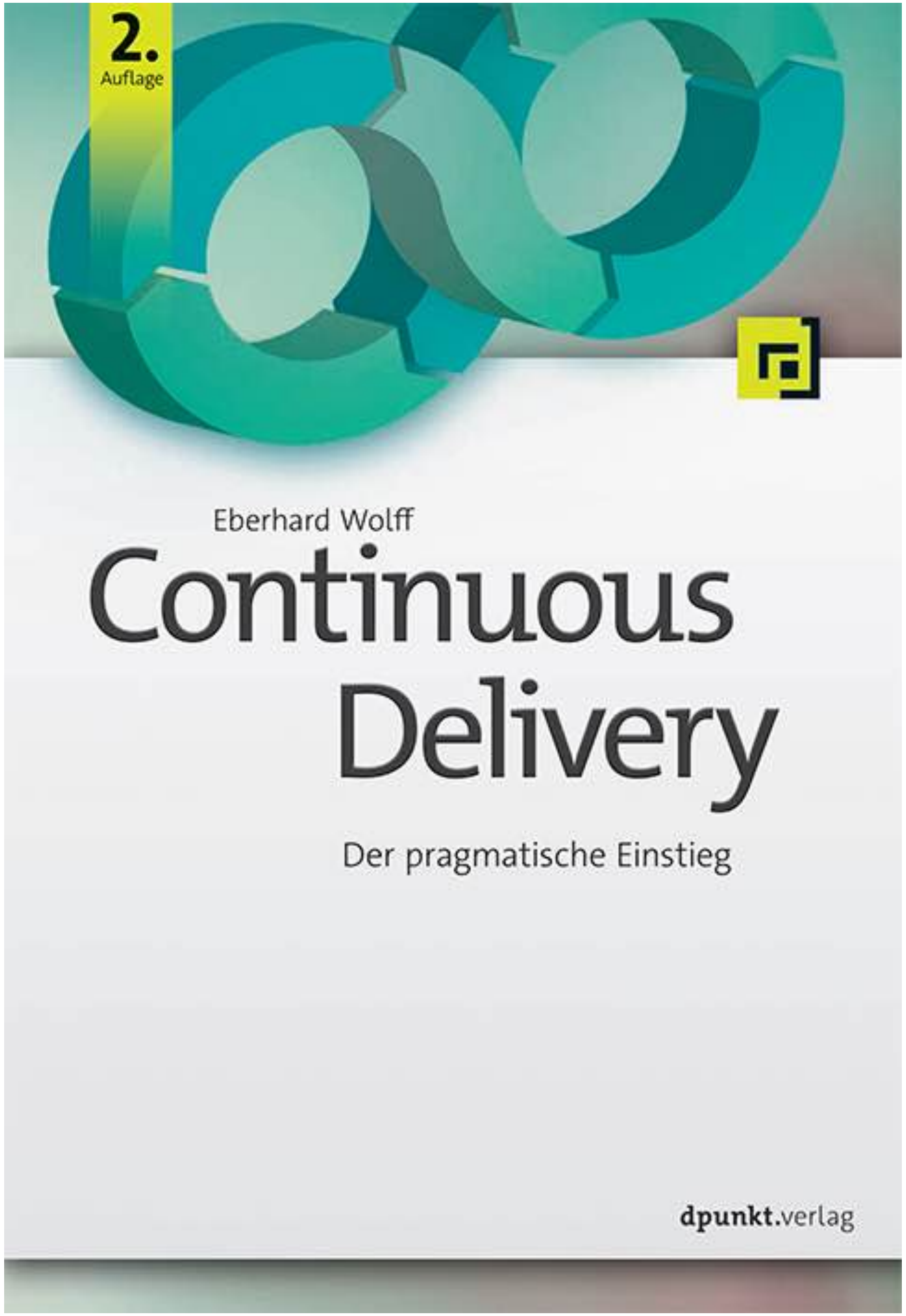
The Frontiers of Continuous Delivery

Eberhard Wolff
@ewolff
http://ewolff.com
Fellow



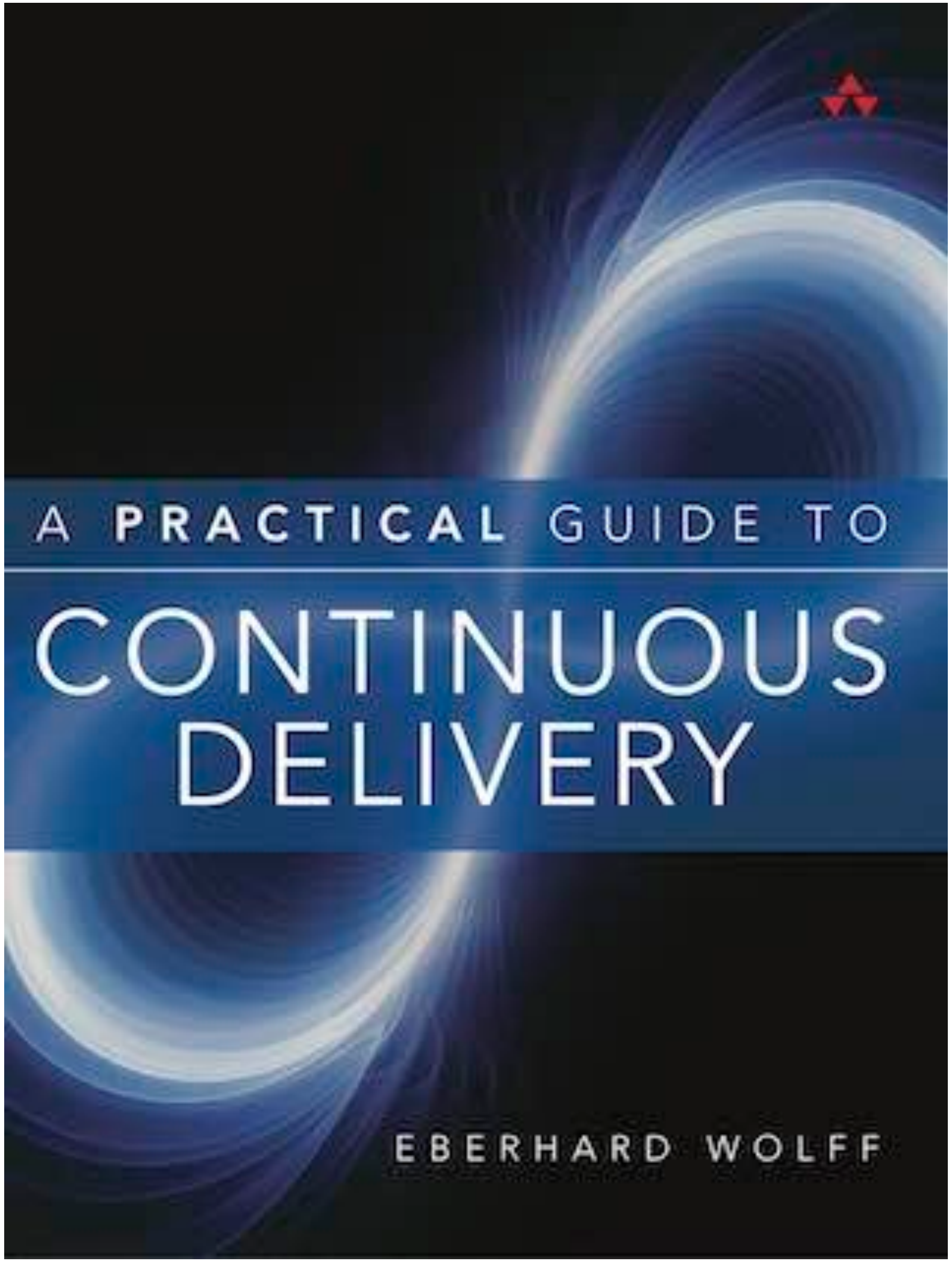
2. Auflage

Eberhard Wolff

Continuous Delivery

Der pragmatische Einstieg

dpunkt.verlag



A PRACTICAL GUIDE TO

CONTINUOUS DELIVERY

EBERHARD WOLFF

http://continuous-delivery-buch.de/

http://continuous-delivery-book.com/

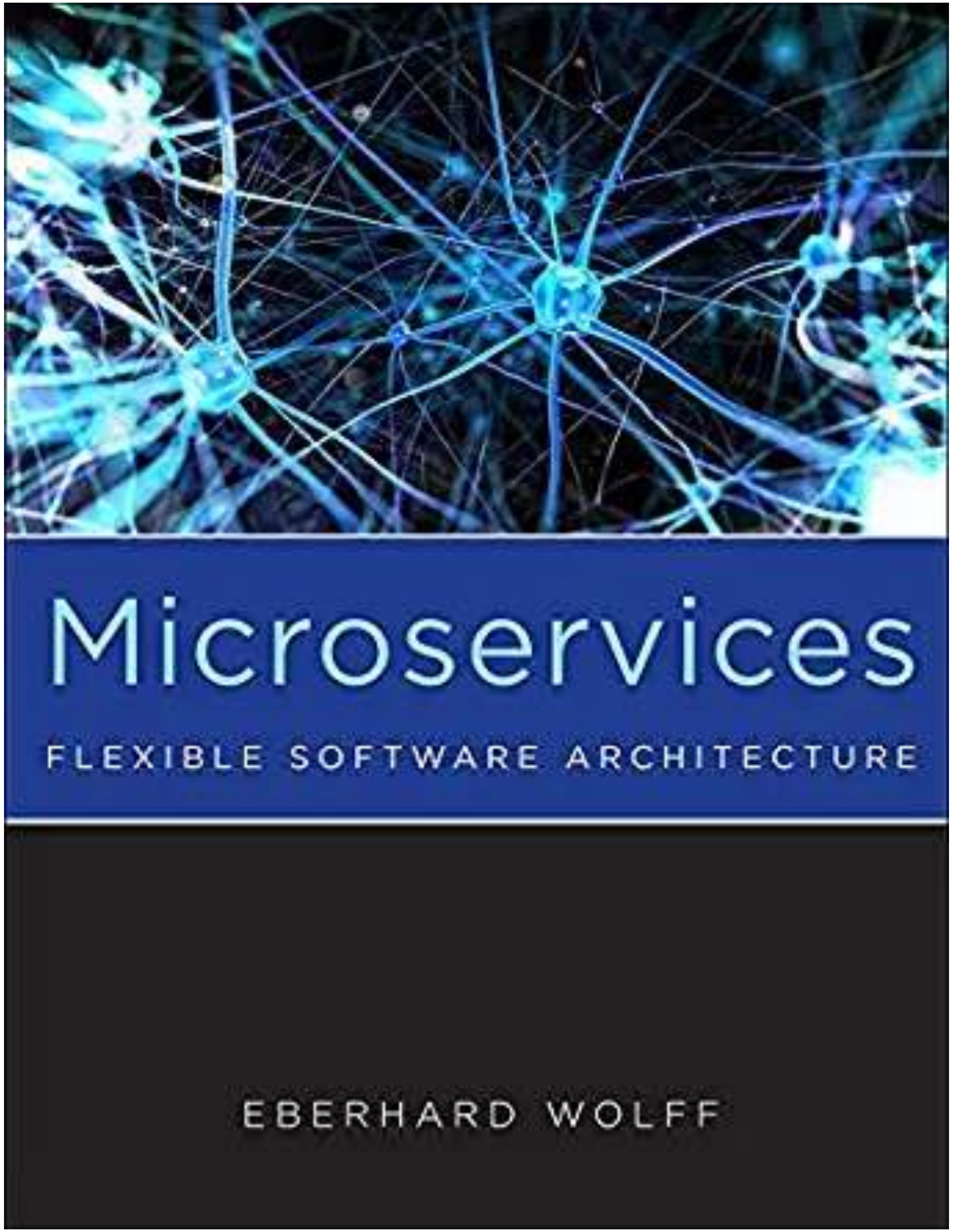
Eberhard Wolff

Microservices

Grundlagen flexibler Softwarearchitekturen

dpunkt.verlag

http://microservices-buch.de/

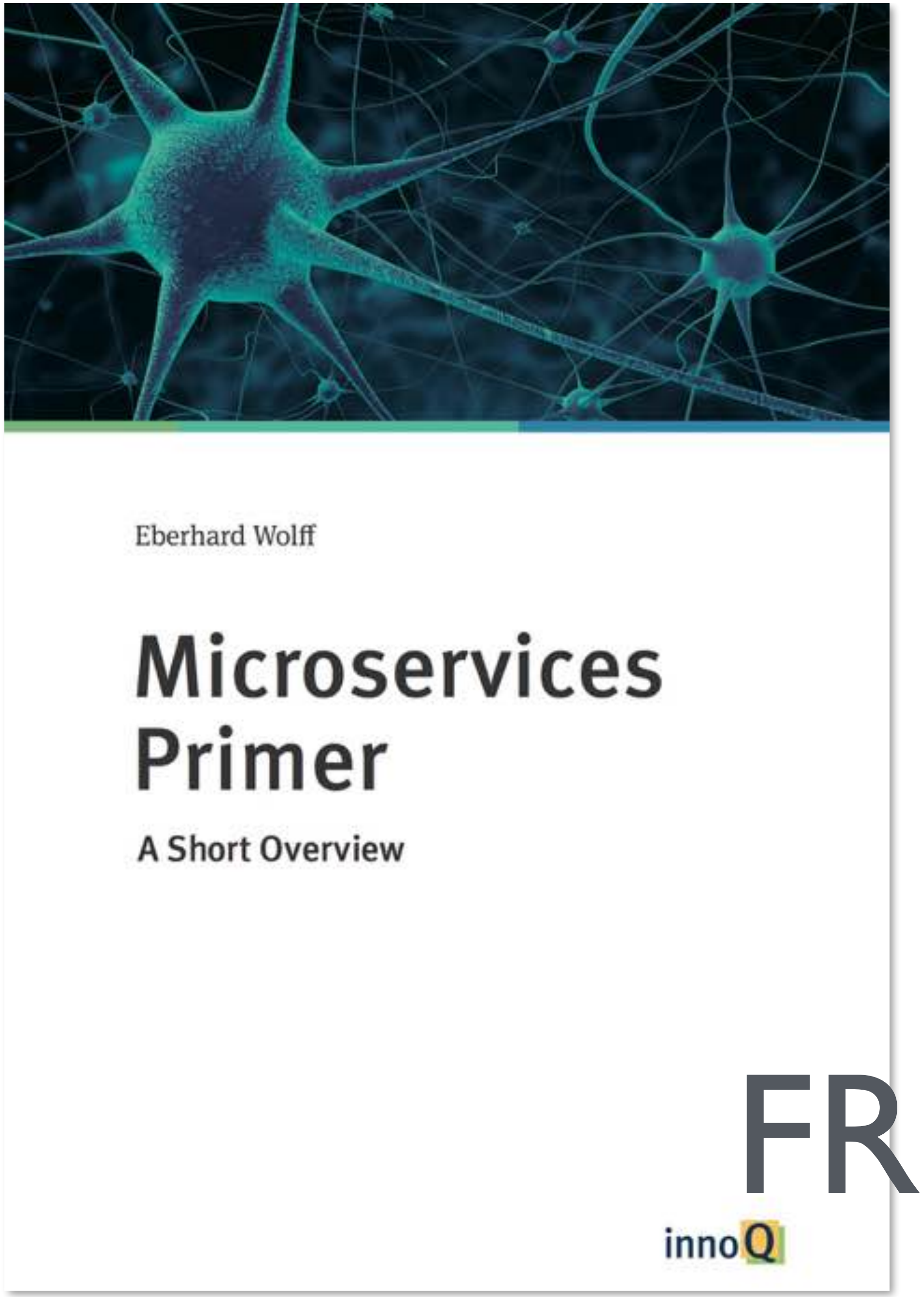


Microservices

FLEXIBLE SOFTWARE ARCHITECTURE

EBERHARD WOLFF

http://microservices-book.com/

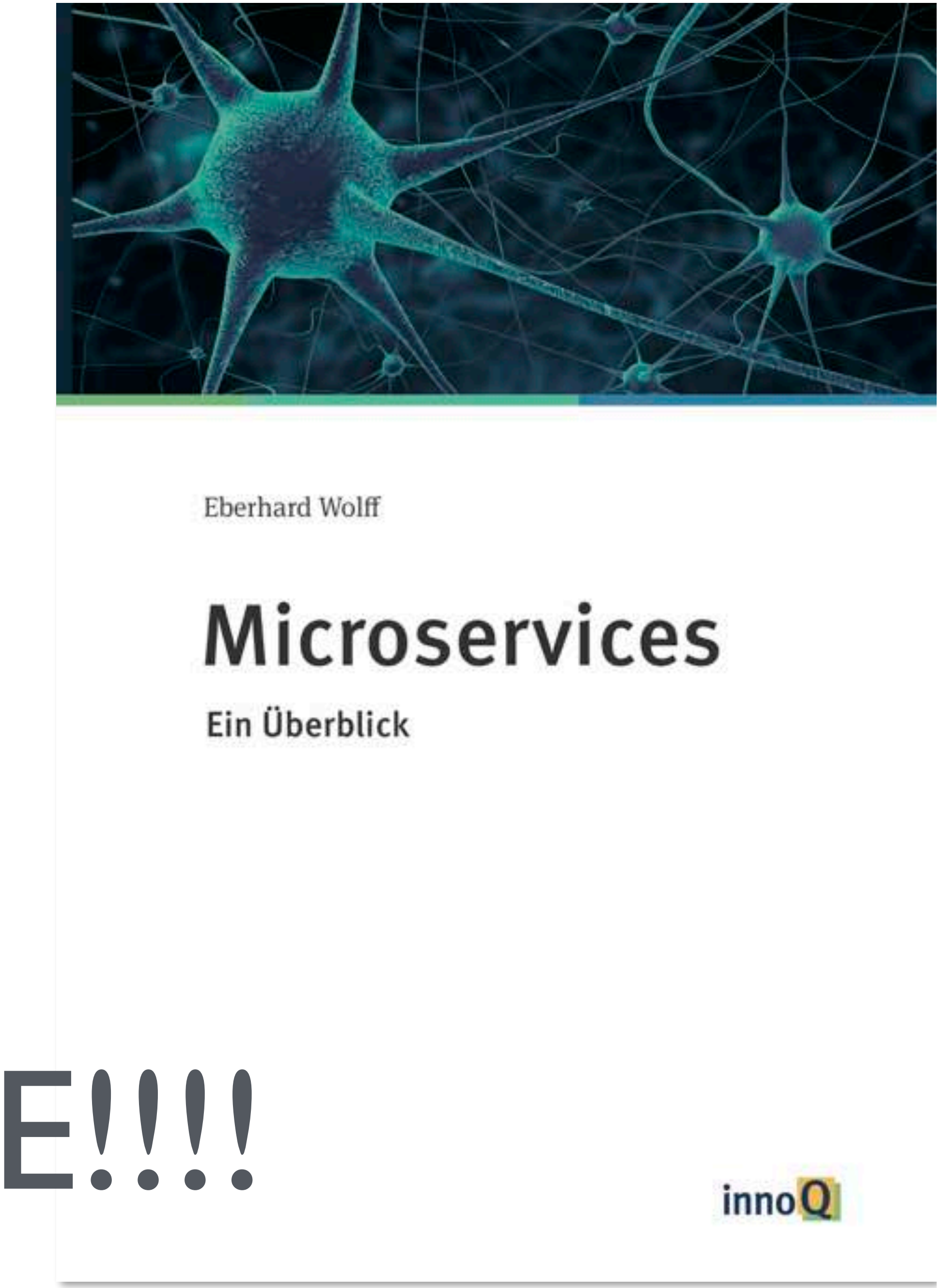


Eberhard Wolff

Microservices Primer

A Short Overview

innoQ



Eberhard Wolff

Microservices

Ein Überblick

innoQ

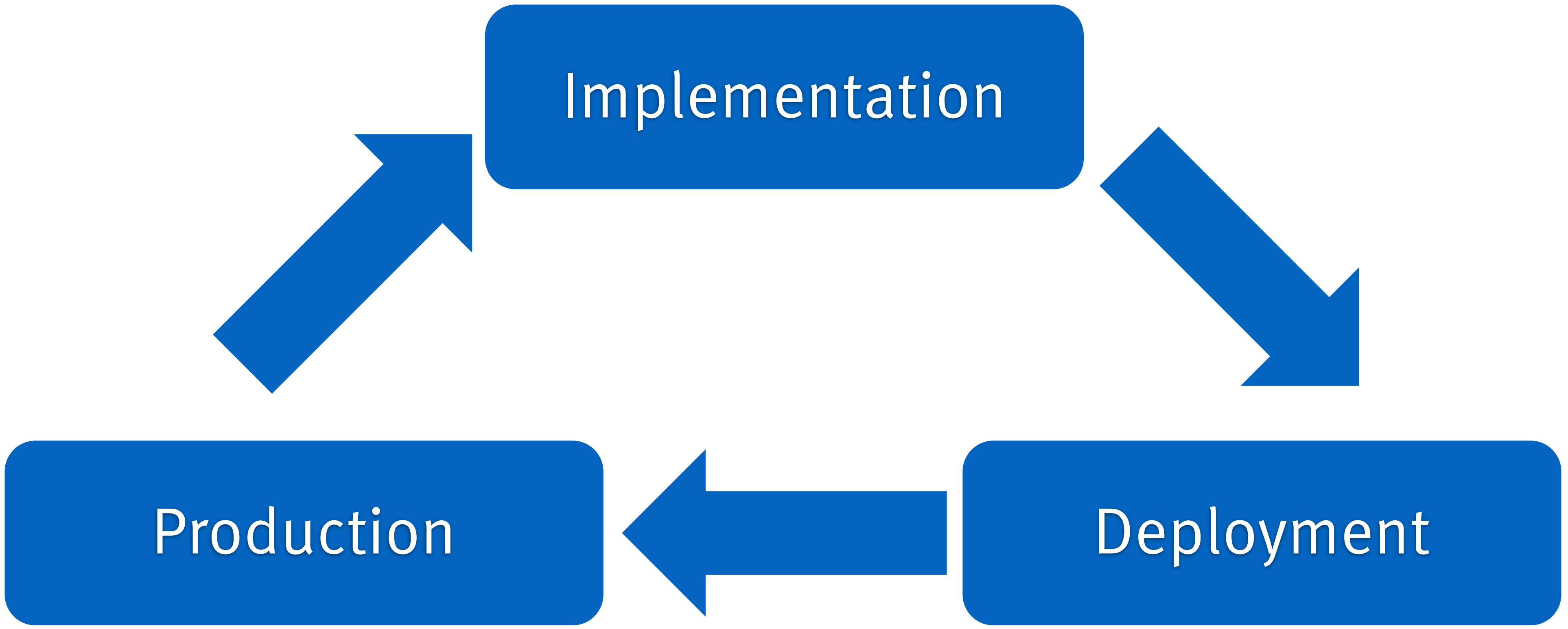
FREE!!!!

http://microservices-book.com/primer.html

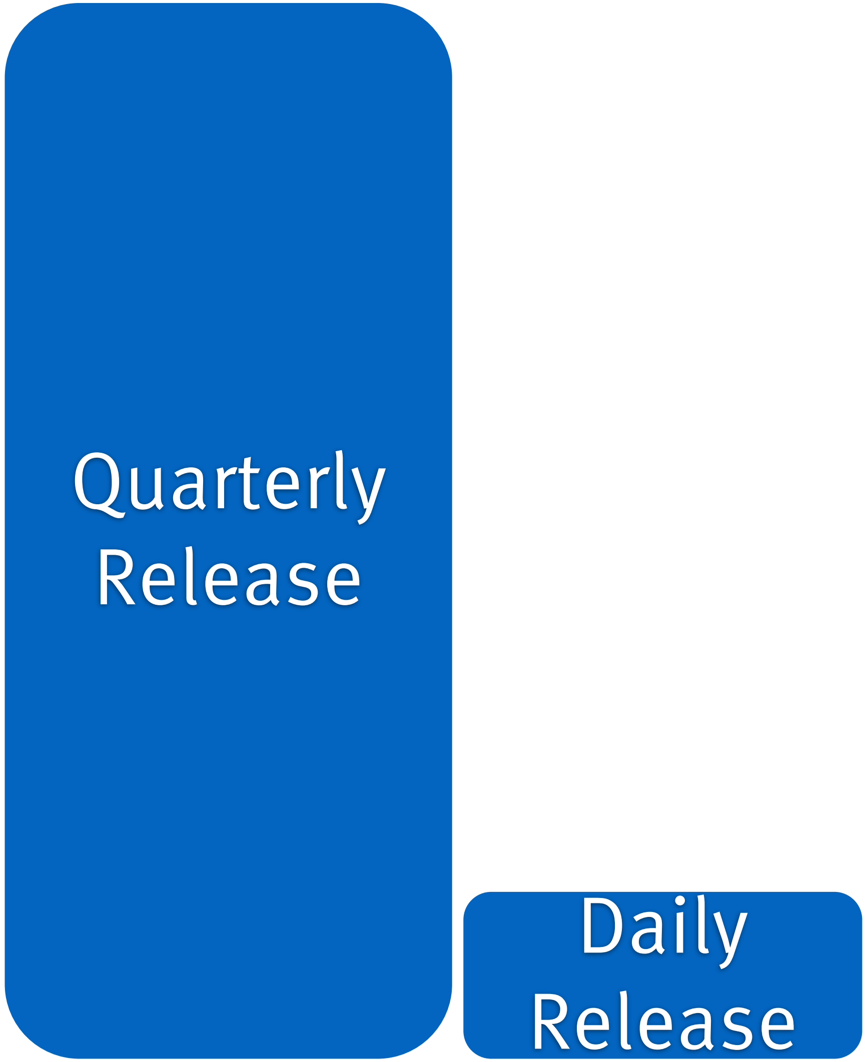
http://microservices-buch.de/ueberblick.html

Continuous Delivery – Why Do I Even Care?

Faster Feedback

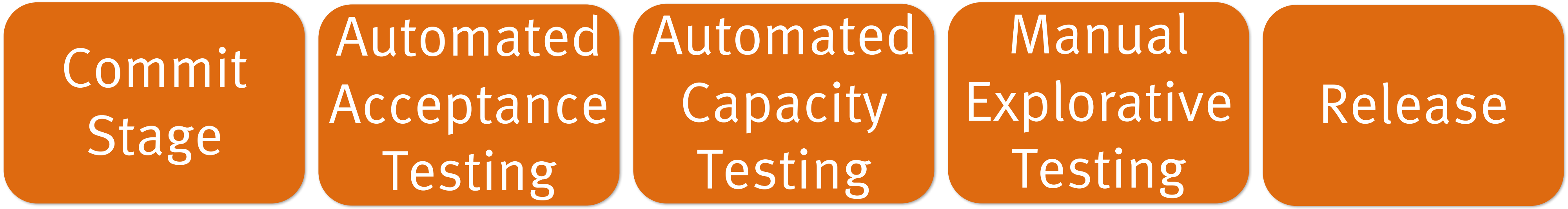


Lower Risk



- Much less deployed
- Less risk of a bug
- Easier to fall back
- ...or add other safeguards

Higher Reliability



Commit Stage

Automated Acceptance Testing

Automated Capacity Testing

Manual Explorative Testing

Release

- Automated deployment and tests
- …easy to reproduce
- …faster
- …executed frequently

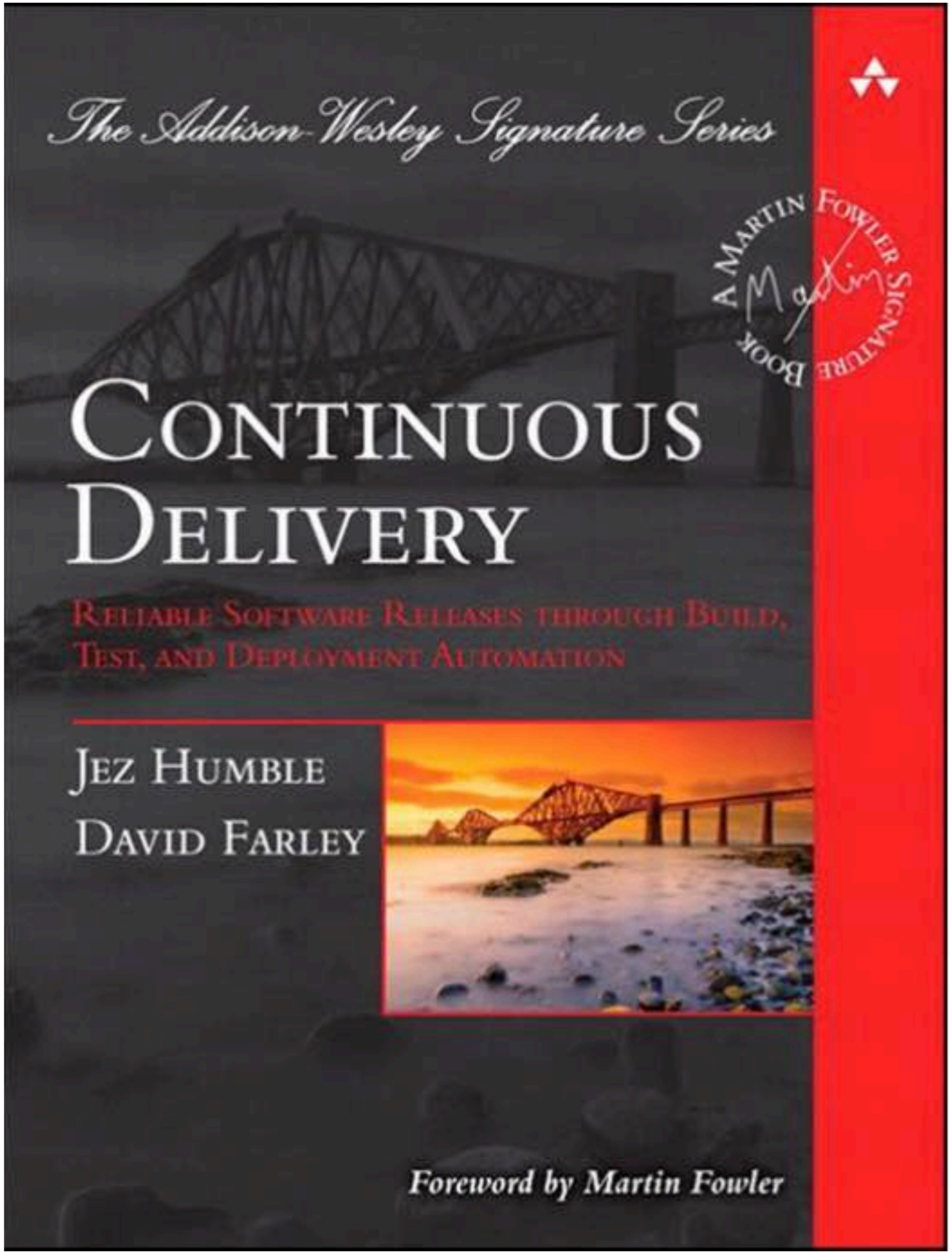
Principles Agile Manifesto

Our highest priority is
to satisfy the customer
through early and
continuous delivery
of valuable software.

Continuous Delivery: Why Do I Even Care?

- Faster Feedback
- Lower Risk
- Higher Reliability
- Value to the customer

- I’m in!


The Addison-Wesley Signature Series
A Martin Fowler Signature Book
Continuous Delivery
Reliable Software Releases through Build, Test, and Deployment Automation
Jez Humble
David Farley
Foreword by Martin Fowler

2010: Continuous Delivery is the next big thing!

Continuous Delivery will increase productivity!

Continuous Delivery should obviously be the way to go.

Continuous Delivery = Technical Issue

Continuous Delivery = Technical Issue Deployment

No

kubernetes

go

amazon web services

2017: Lots of tools to solve technical issues.

CHEF™

puppet

docker

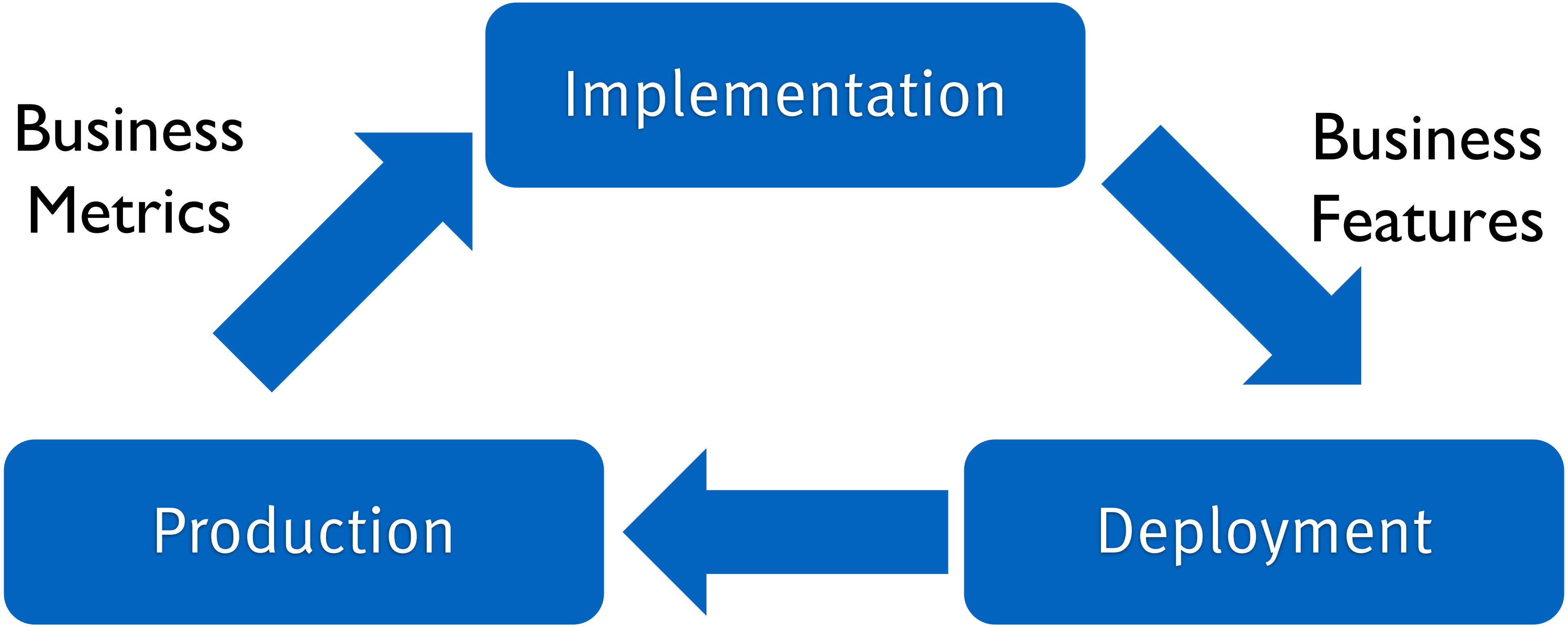
Continuous Delivery is People.

People Continuous Delivery Business

Management
Buy-In

Frontier: Business

Faster Feedback



Implementation

Business Features

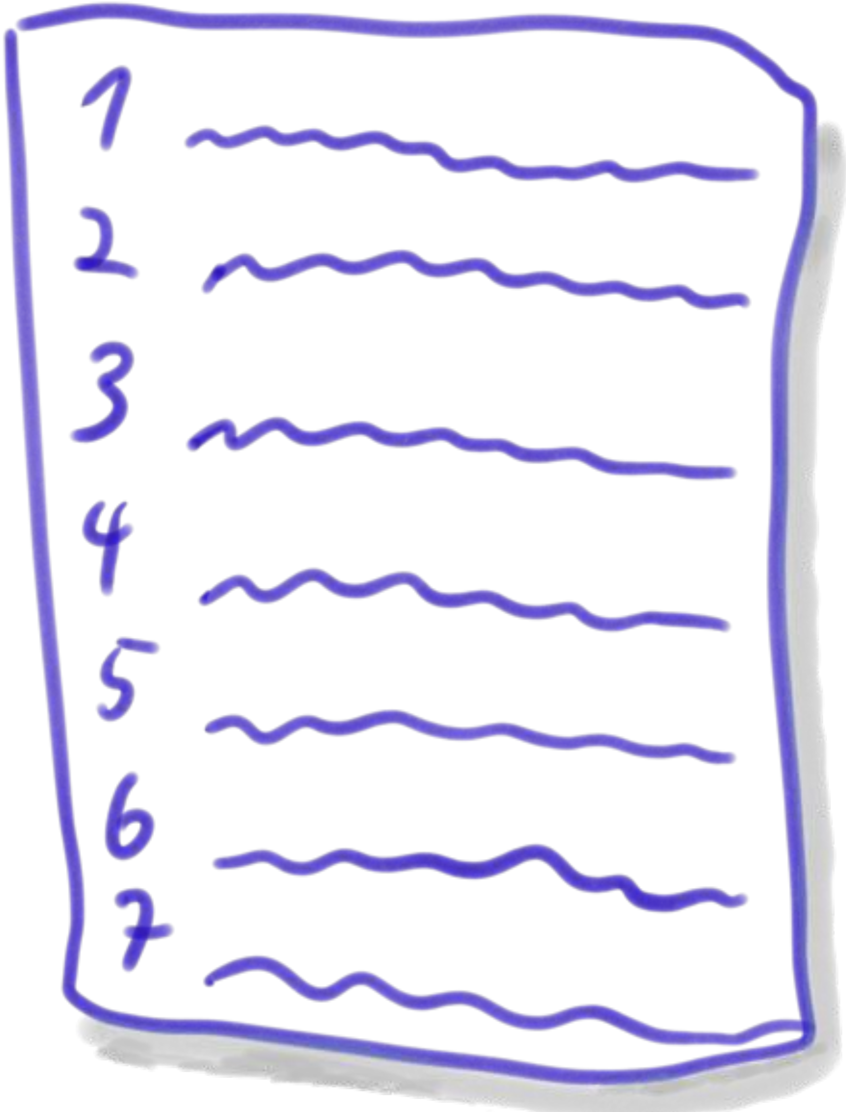
Deployment

Production

Business Metrics

How Business Works

- Release Q1/2018
- Here are the features!
- Go!

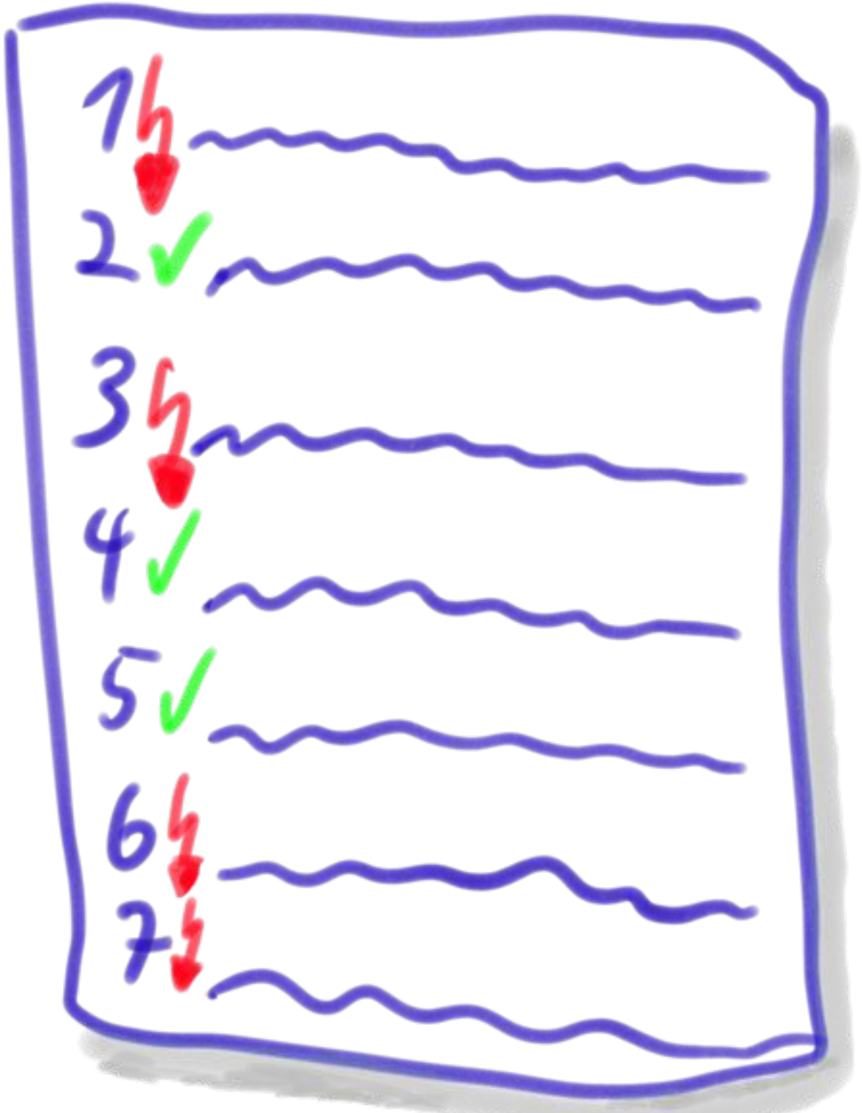


60%– 90% of ideas do not improve the metrics they were intended to improve

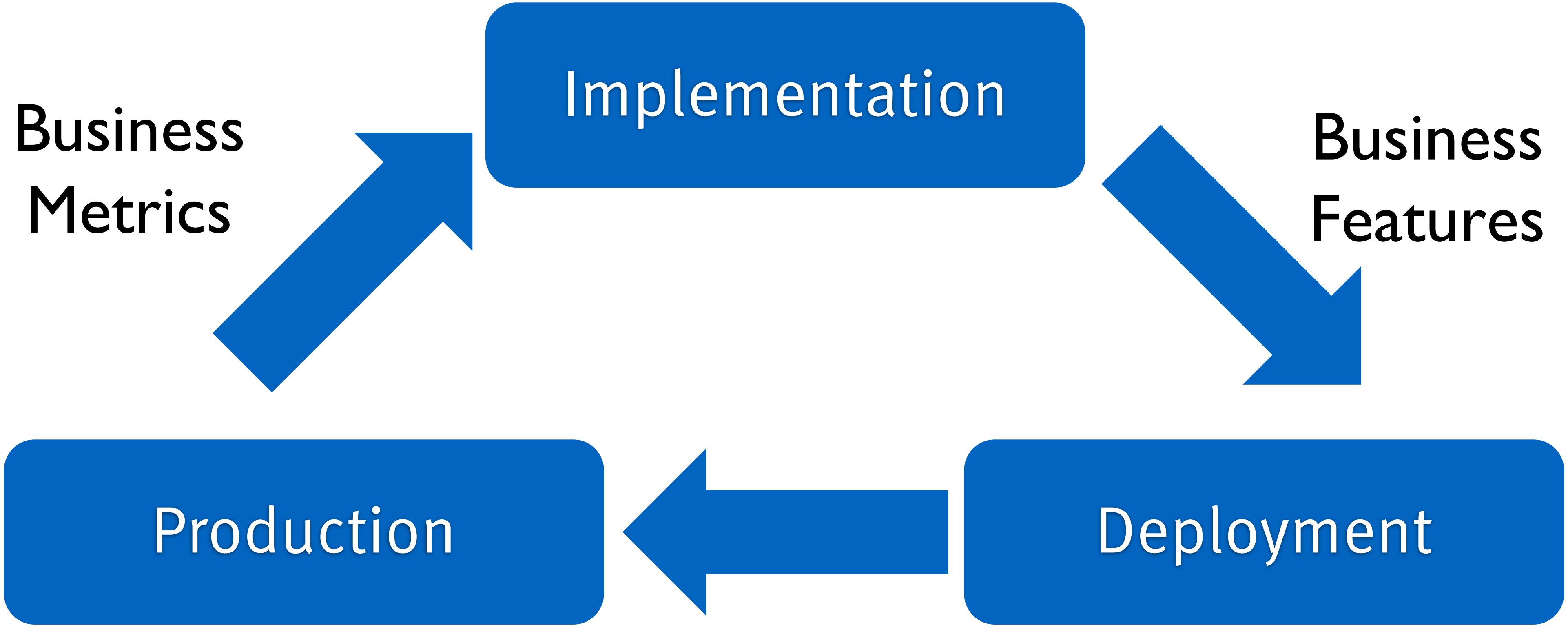
Ronny Kohavi
Former Head Data Mining and Personalization group
Amazon
Source: Lean Enterprise, Humble et al

Just Waste

- More than half of the features are worthless…
- …or hurt business goals.
- Many businesses doesn’t even know the KPIs.



Run a minimal feature by users.



Related to MVP (Minimal Viable Product)

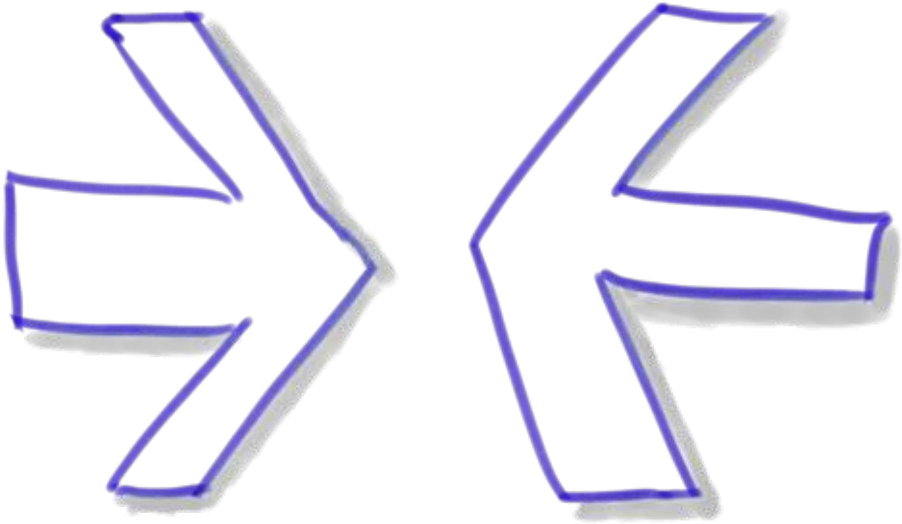
Survival is Optional.

We were too slow

- Fast releases lead to better software and products.
- Bad products die out.
- Continuous delivery: The only way to succeed for a business.

IT
Chauvinism

Ways to Compete



- More features faster
- ...or...
- Trust
- Existing customer relations

- Would your grandpa choose a FinTech over a bank?

Continuous Delivery: No

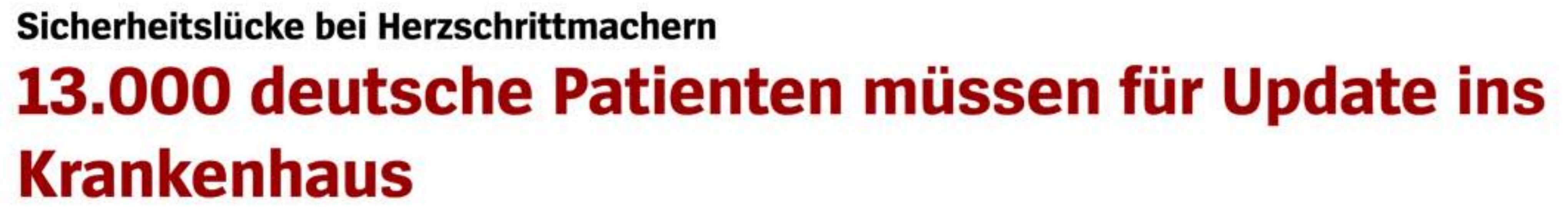
- Diesel update at VW and Audi
- 4.000.000 cars going to the garage just for a software update.
- How much does that cost?
- Per car 70€
- Total 280.000.000€

https://heise.de/newsticker/meldung/Volkswagen-Haendler-Software-Update-taugt-nicht-3834343.html

http://www.handelsblatt.com/my/unternehmen/industrie/volkswagen-vier-millionen-diesel-autos-erhalten-update/20139344.html

Continuous Delivery: Yes

- Tesla
- New features like
- ...more speed
- ...more range during hurricane Irma
- ...self-driving
- ...summoning
- etc..



Sicherheitslücke bei Herzschrittmachern

13.000 deutsche Patienten müssen für Update ins Krankenhaus

http://spon.de/ae3nr

Even for cars

most features are

software now.

Even for cars
you can do
continuous delivery.

So you really don’t see any value?

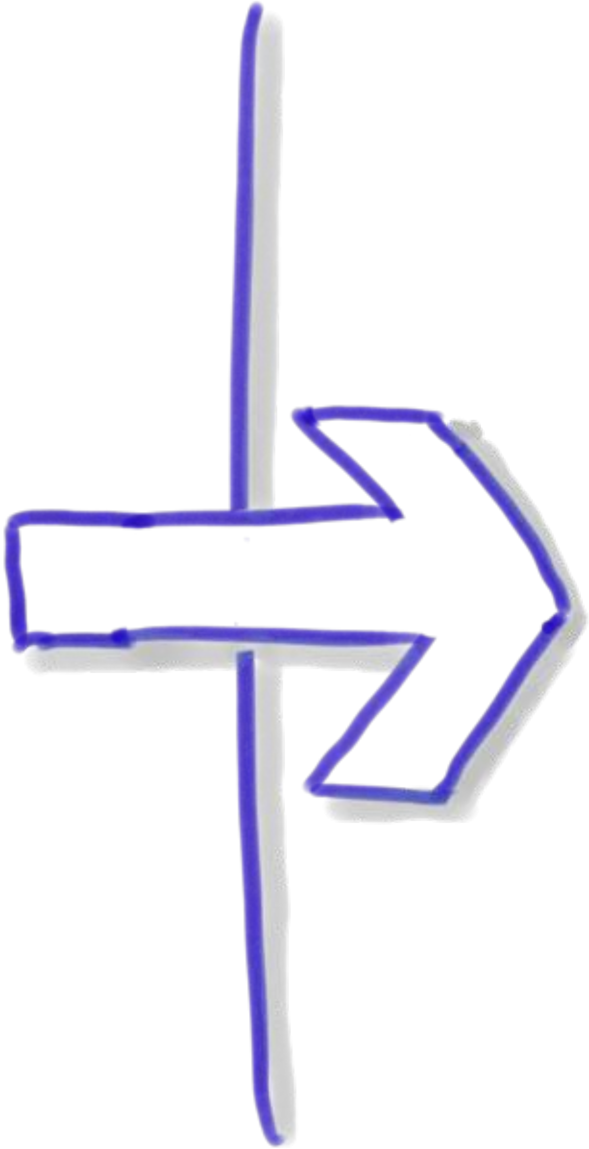
You really can’t do Continuous Delivery?

Business

- ...could get the biggest benefit
- ...but often doesn‘t

- Product development in small batches is different from the known ways...
- ...and some businesses are not under a lot of pressure.

Extending the frontier



Is Continuous Delivery worth it without business support?

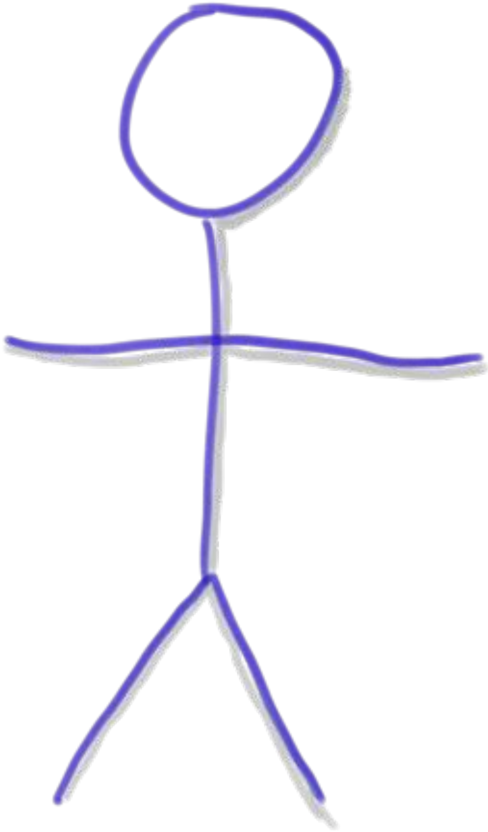
YES!!

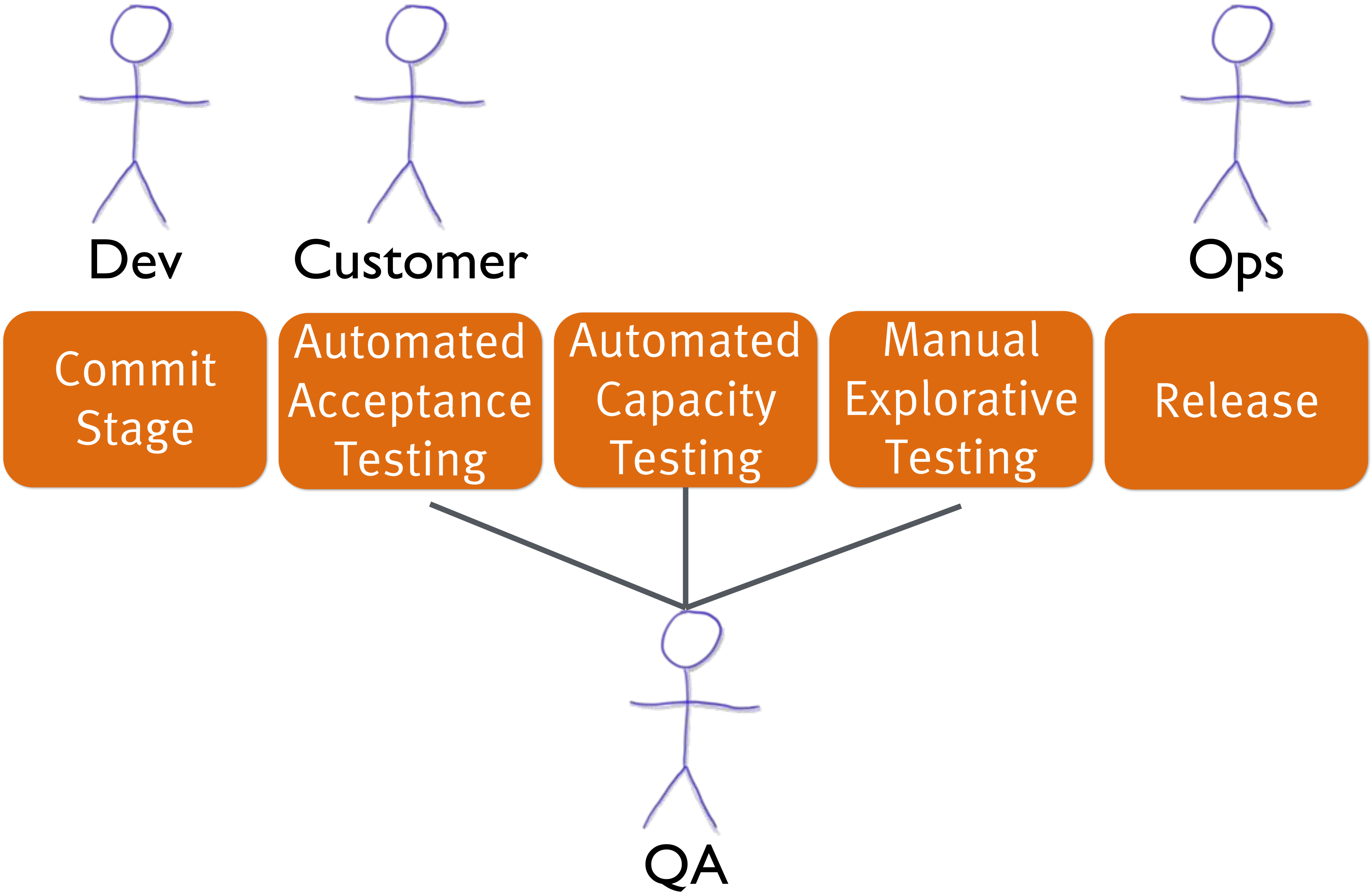
- Faster Feedback
- Lower Risk
- Higher Reliability
- ~~Value to the customer~~

Extending the Business Frontier

- Ambitious: IT drives the business
- Not too much influence?
- IT sometimes only think they know better
- Educate business
- …or focus on other values

Frontier: People





Dev

Customer

Ops

Commit Stage

Automated Acceptance Testing

Automated Capacity Testing

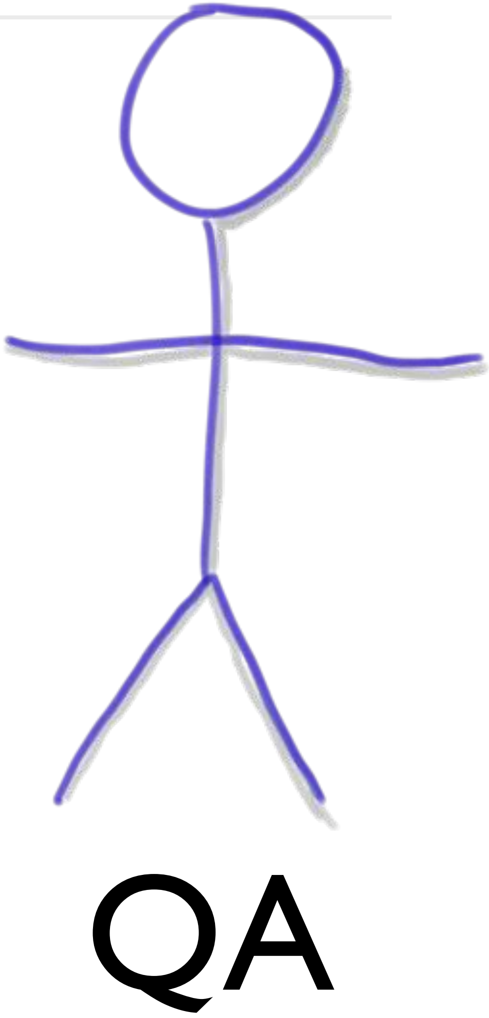
Manual Explorative Testing

Release

QA

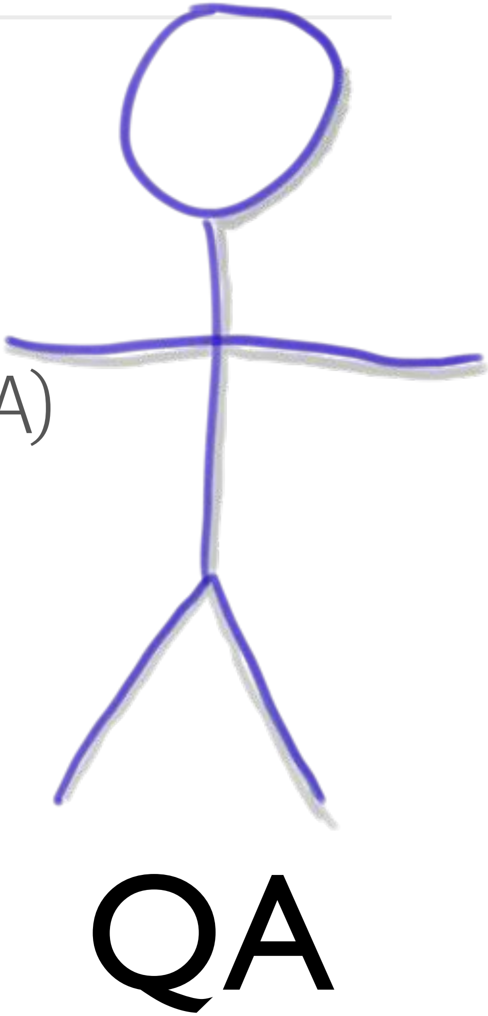
QA & CD

- Quality Assurance (QA) must provide tests
- ...or at least support testing
- Automated tests
- Manual tests too slow
- ...and too error prone



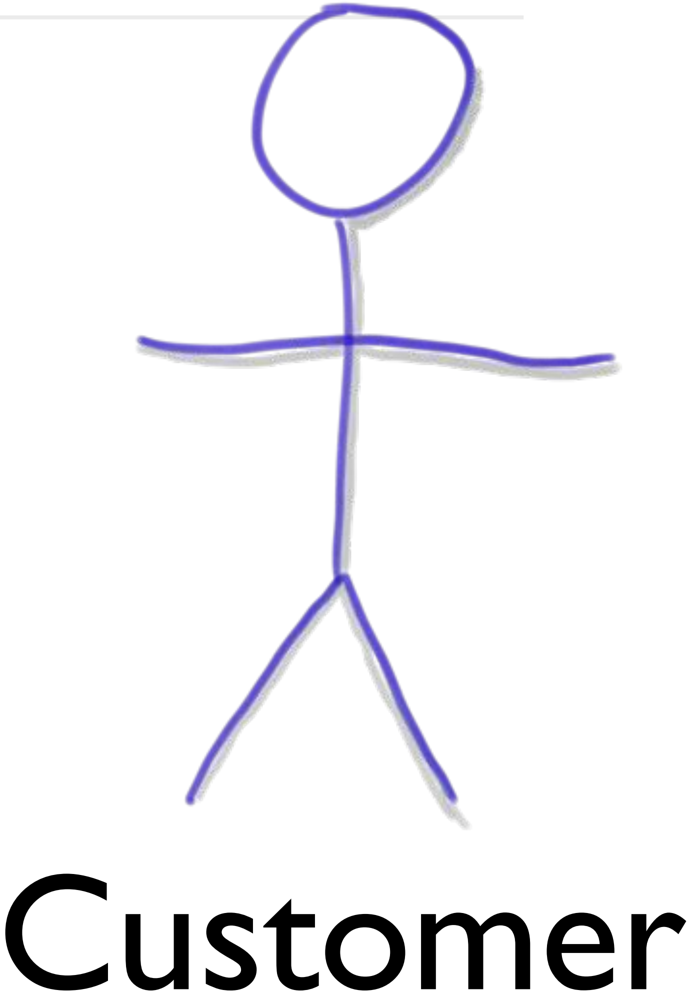
QA & CD

- Traditional Quality Assurance (QA) focuses on manual tests.
- Mind shift
- ...and different skills



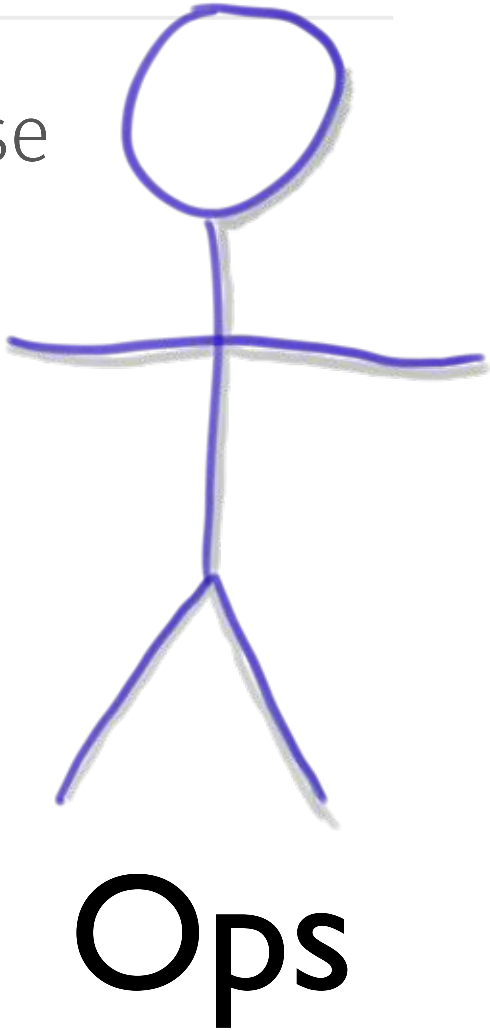
Customer

- Customer must provide information for automated acceptance test
- No more manual sign-off
- Needs trust
- ...and trust!
- ...and some technical literacy



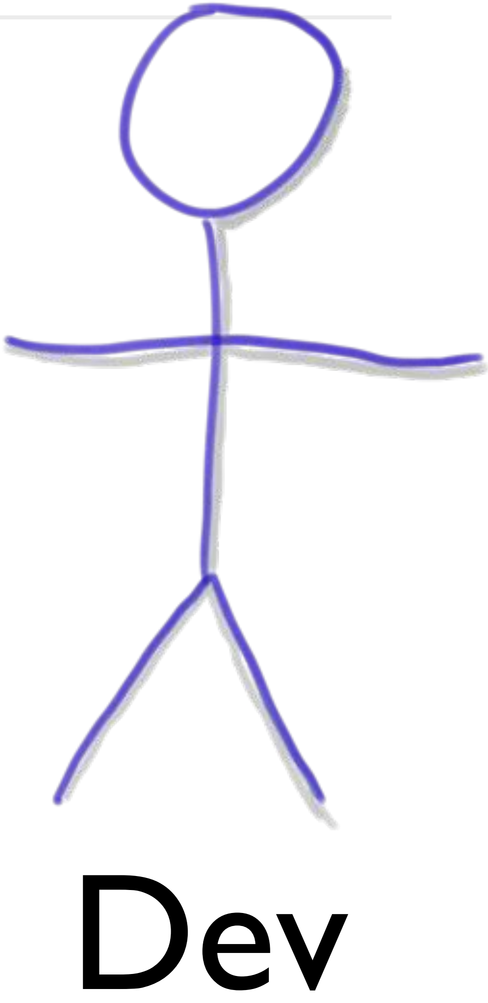
Ops

- One month waiting for a database
- ...that is cheaply provided by a highly optimized Ops team
- ...for "cost"
- Ops has very different incentives
- ...and doesn‘t even work in projects.



Dev

- Can automate
- i.e. develop software
- ...but have limited knowledge about the other topics.

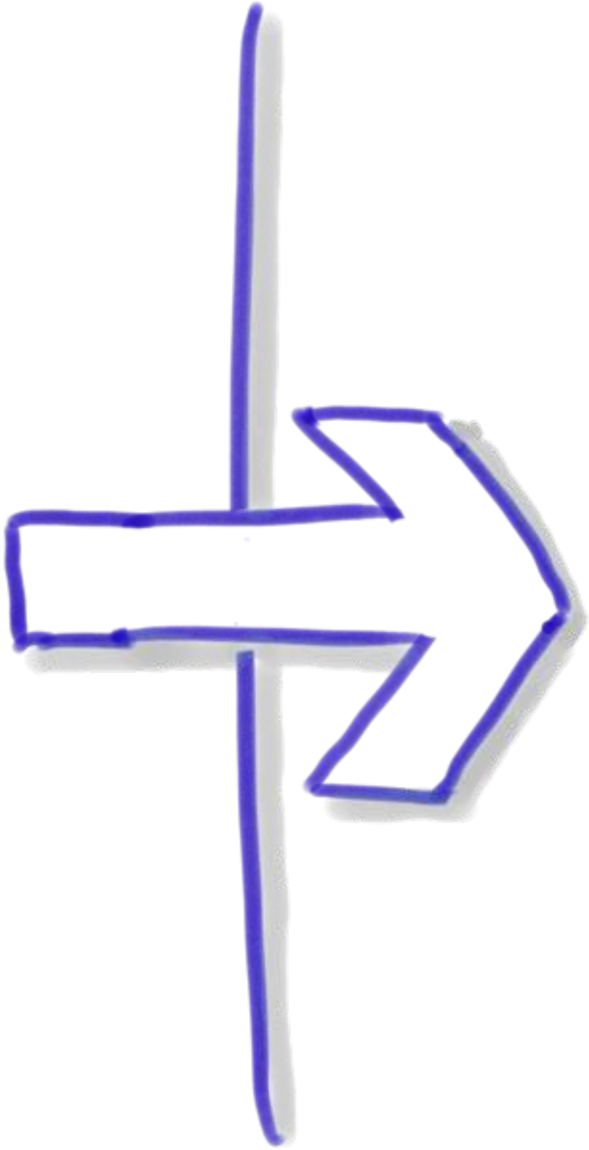


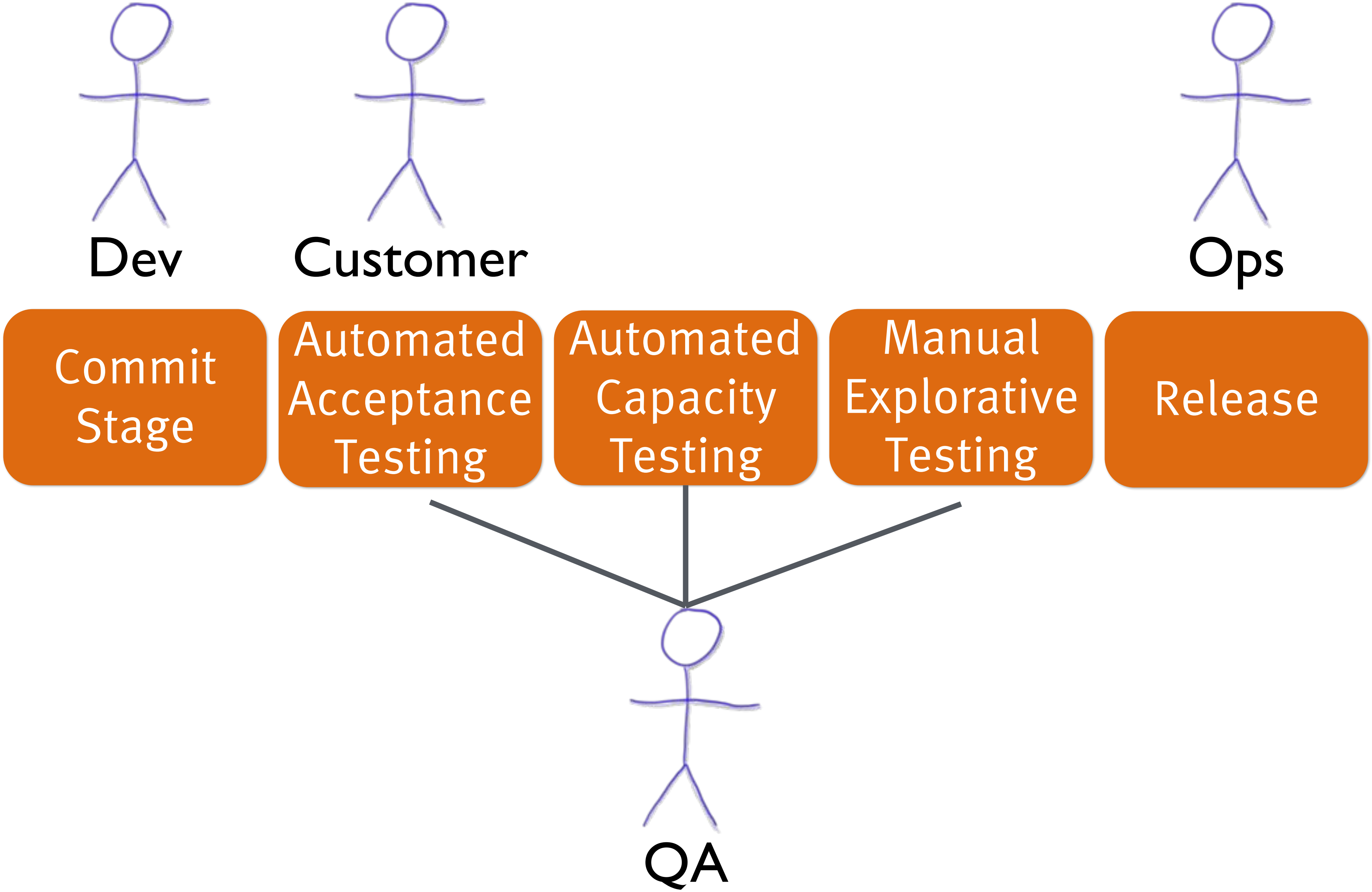
Software = Automation

Software = Automation

Still automating CD is hard!

Extending the frontier





Dev

Customer

Ops

Commit Stage

Automated Acceptance Testing

Automated Capacity Testing

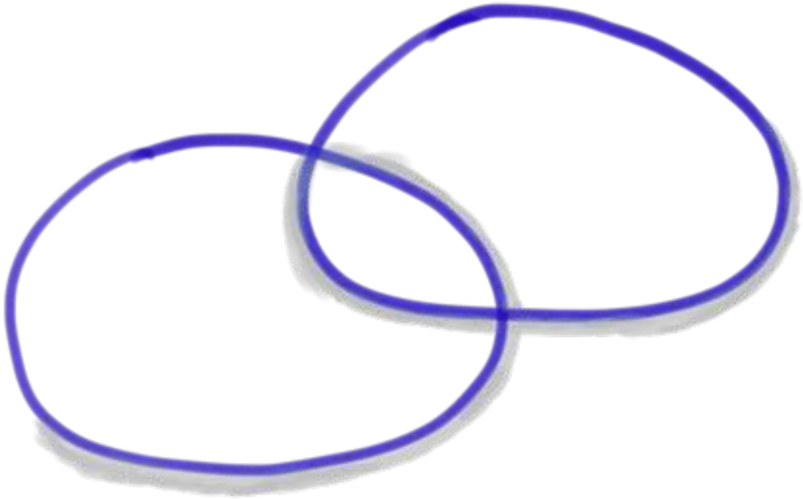
Manual Explorative Testing

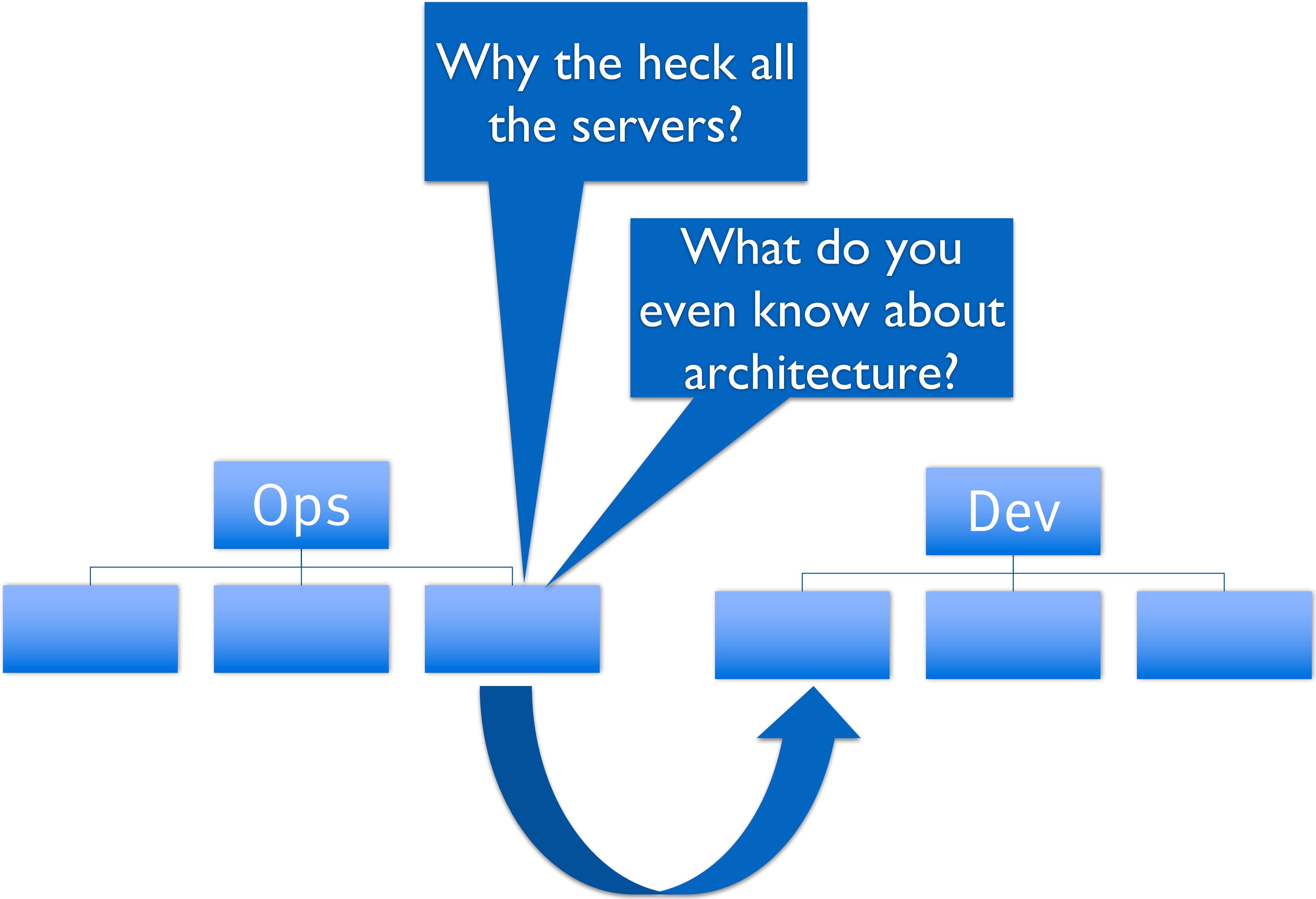
Release

QA

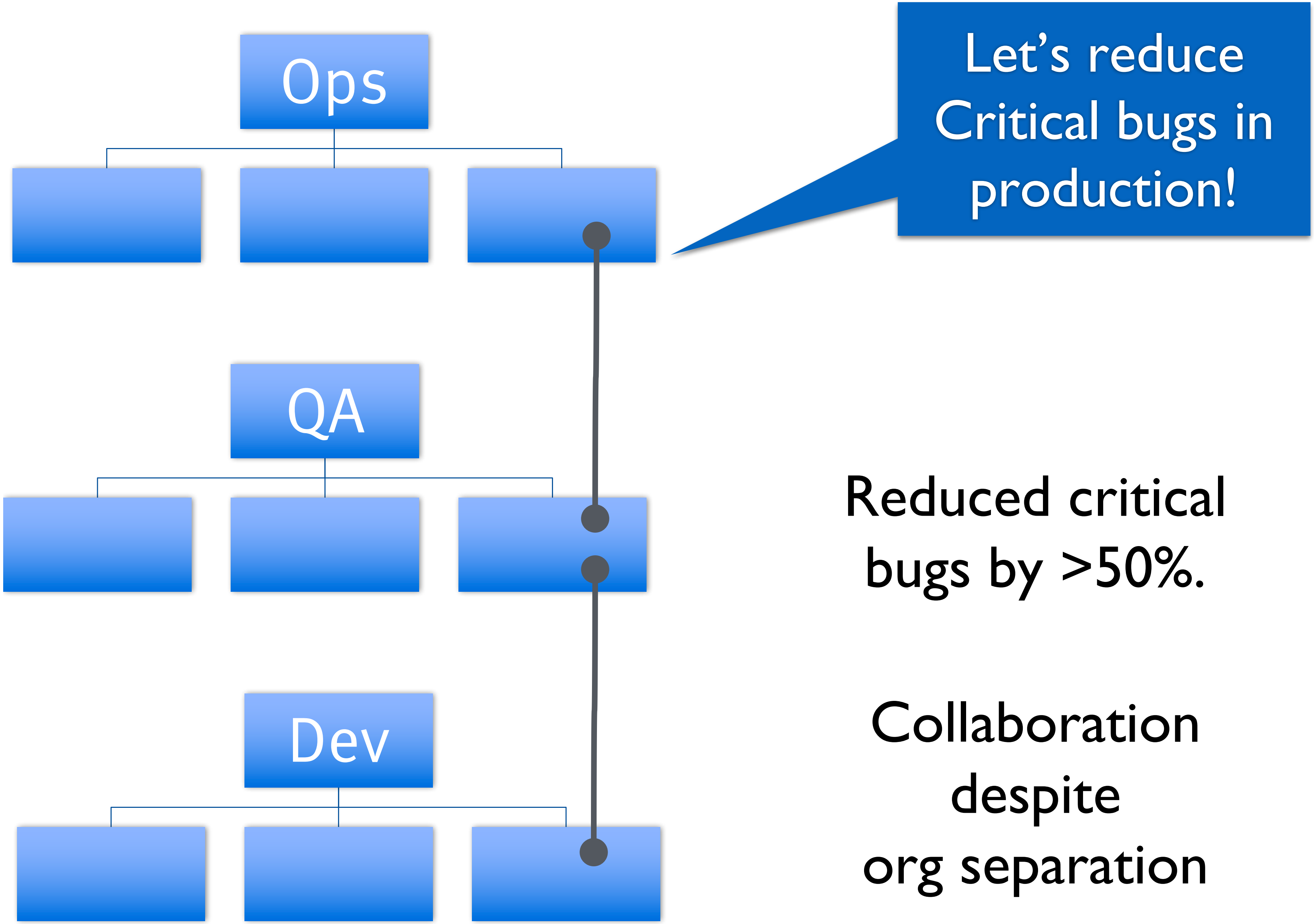
Educate & Collaborate

- Dev do automation all day.
- Make all aware of the needed collaboration
- Encourage collaboration
- Not necessarily an org chart change




Why the heck all the servers?
What do you even know about architecture?
Ops
Dev

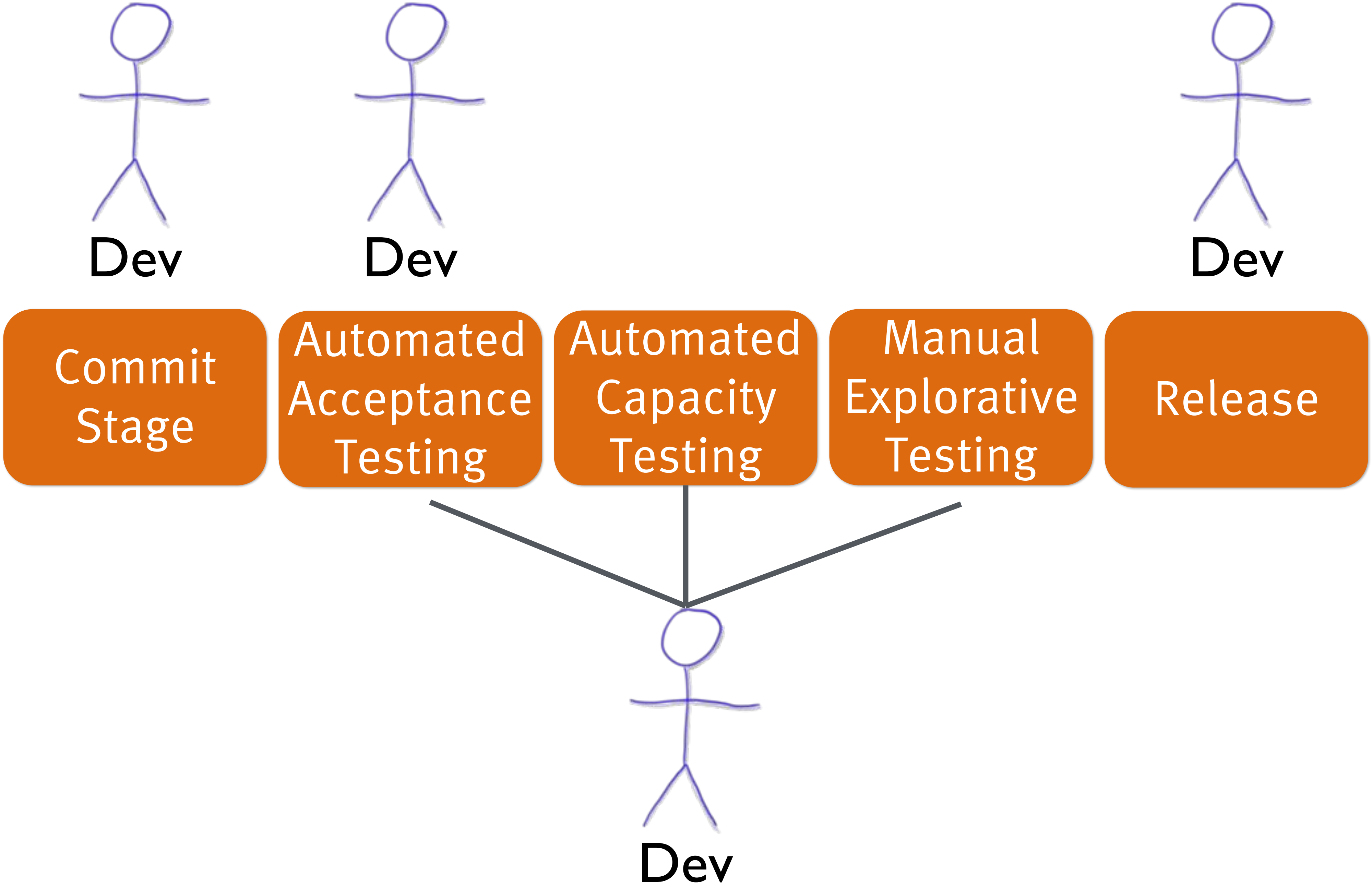
Is reorganization really the solution?



Let's reduce Critical bugs in production!

Reduced critical bugs by >50%.

Collaboration despite org separation



Dev

Dev

Dev

Commit Stage

Automated Acceptance Testing

Automated Capacity Testing

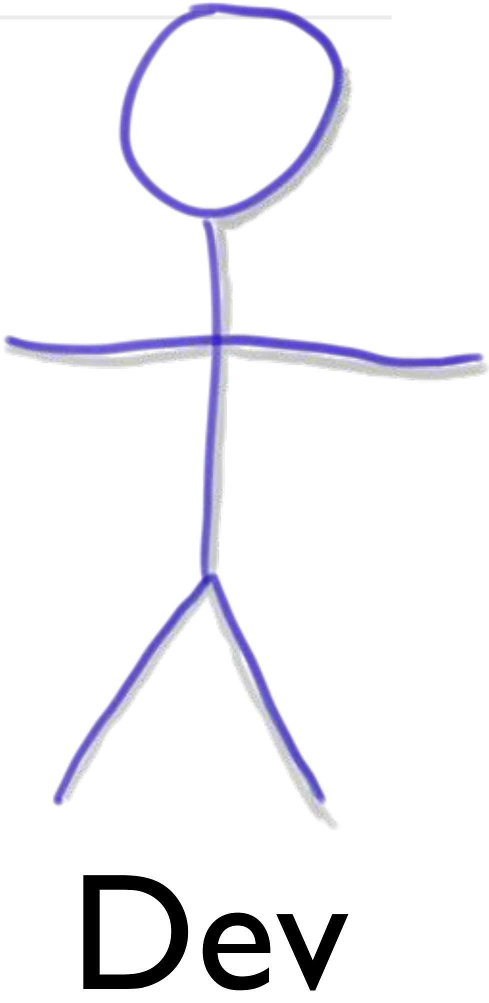
Manual Explorative Testing

Release

Dev

Dev

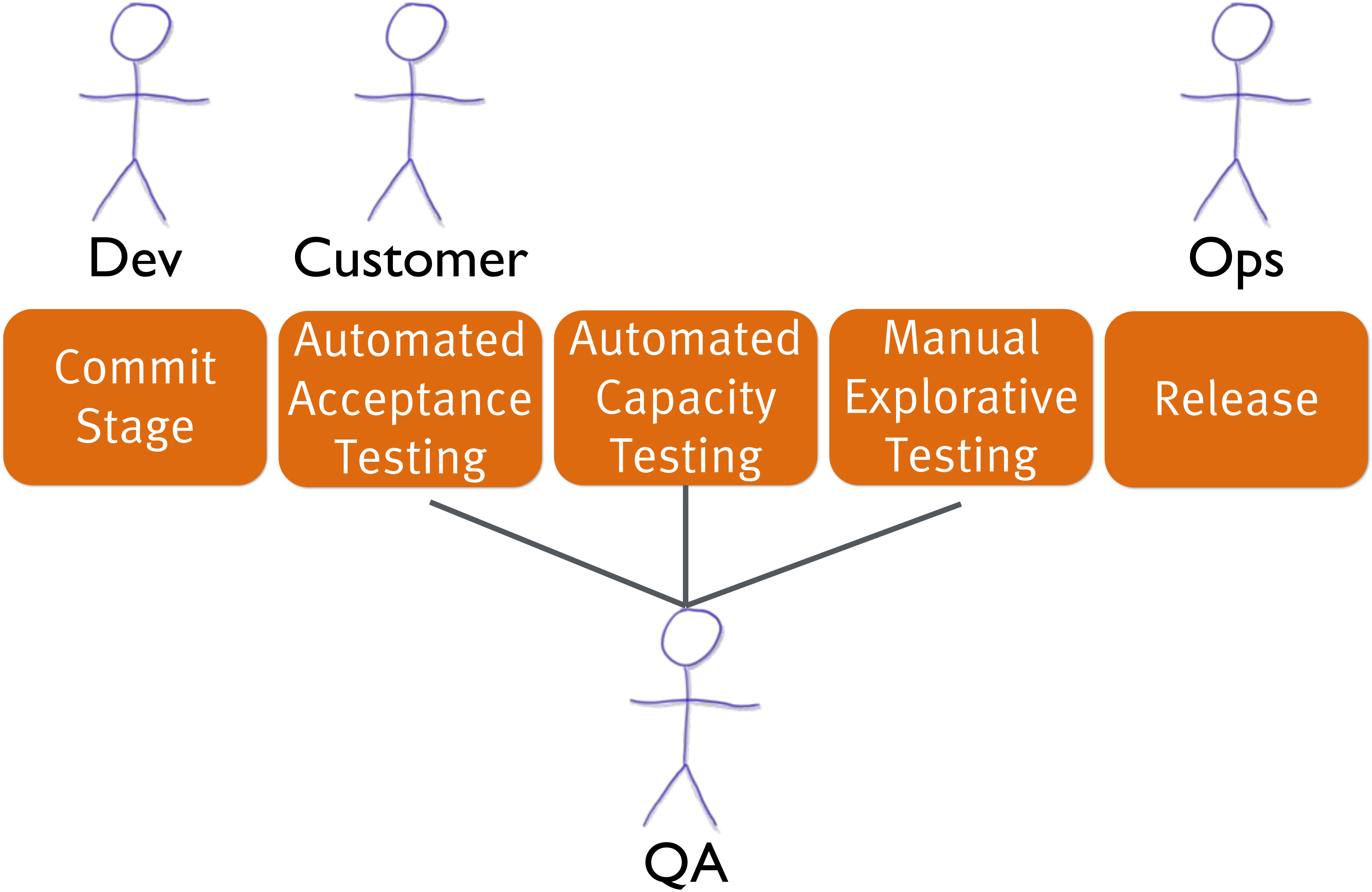
- Dev takes over the other roles.
- Happening in practice
- ...but not a strategy
- Unused QA / Ops skills



2012: Talk about Linux namespaces, AuFS and cgroups at a developer conference?

2017: Docker at every developer conference

Dev is learning Ops skills.



Dev

Customer

Ops

Commit Stage

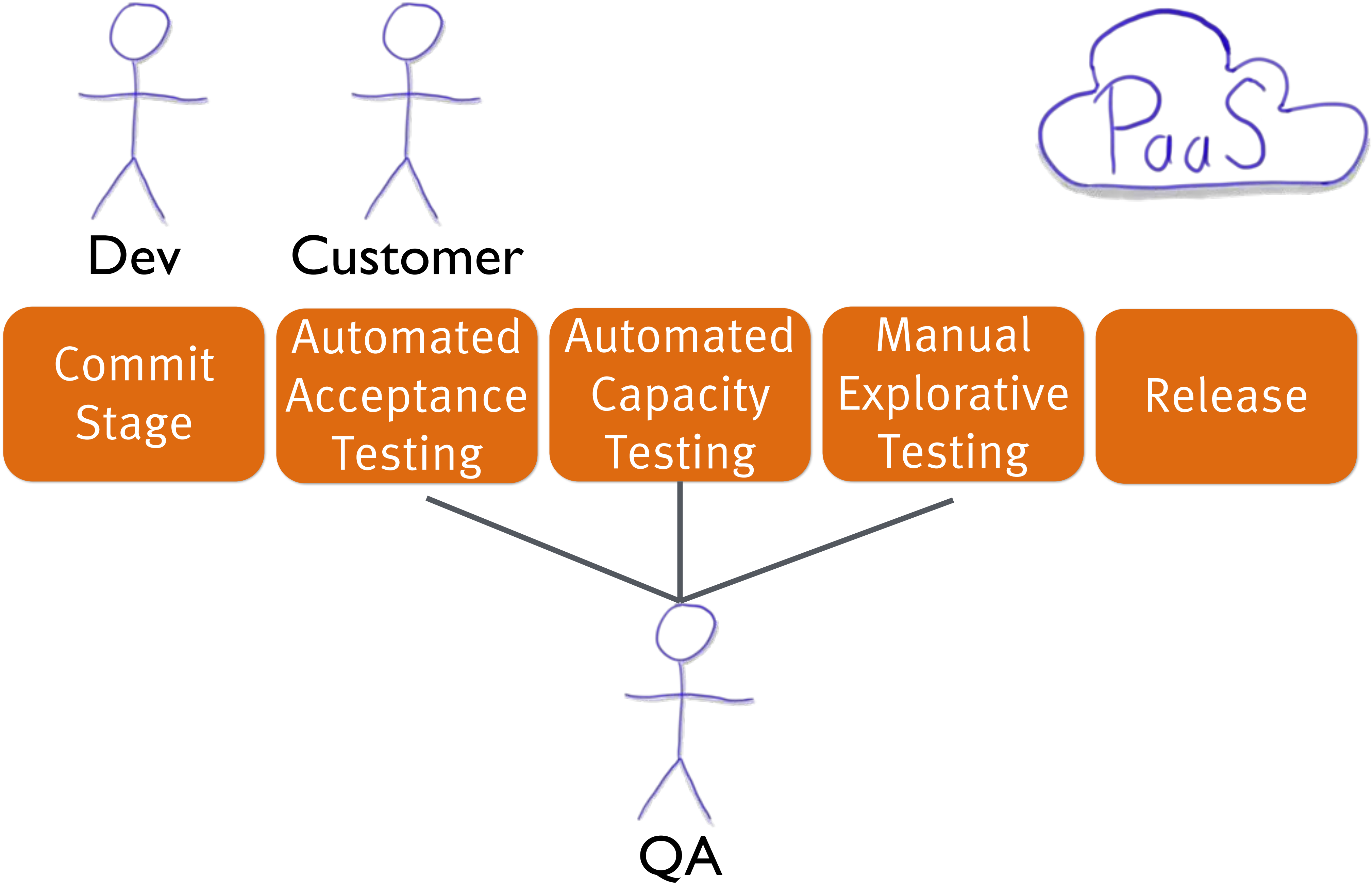
Automated Acceptance Testing

Automated Capacity Testing

Manual Explorative Testing

Release

QA



Dev

Customer

Commit Stage

Automated Acceptance Testing

Automated Capacity Testing

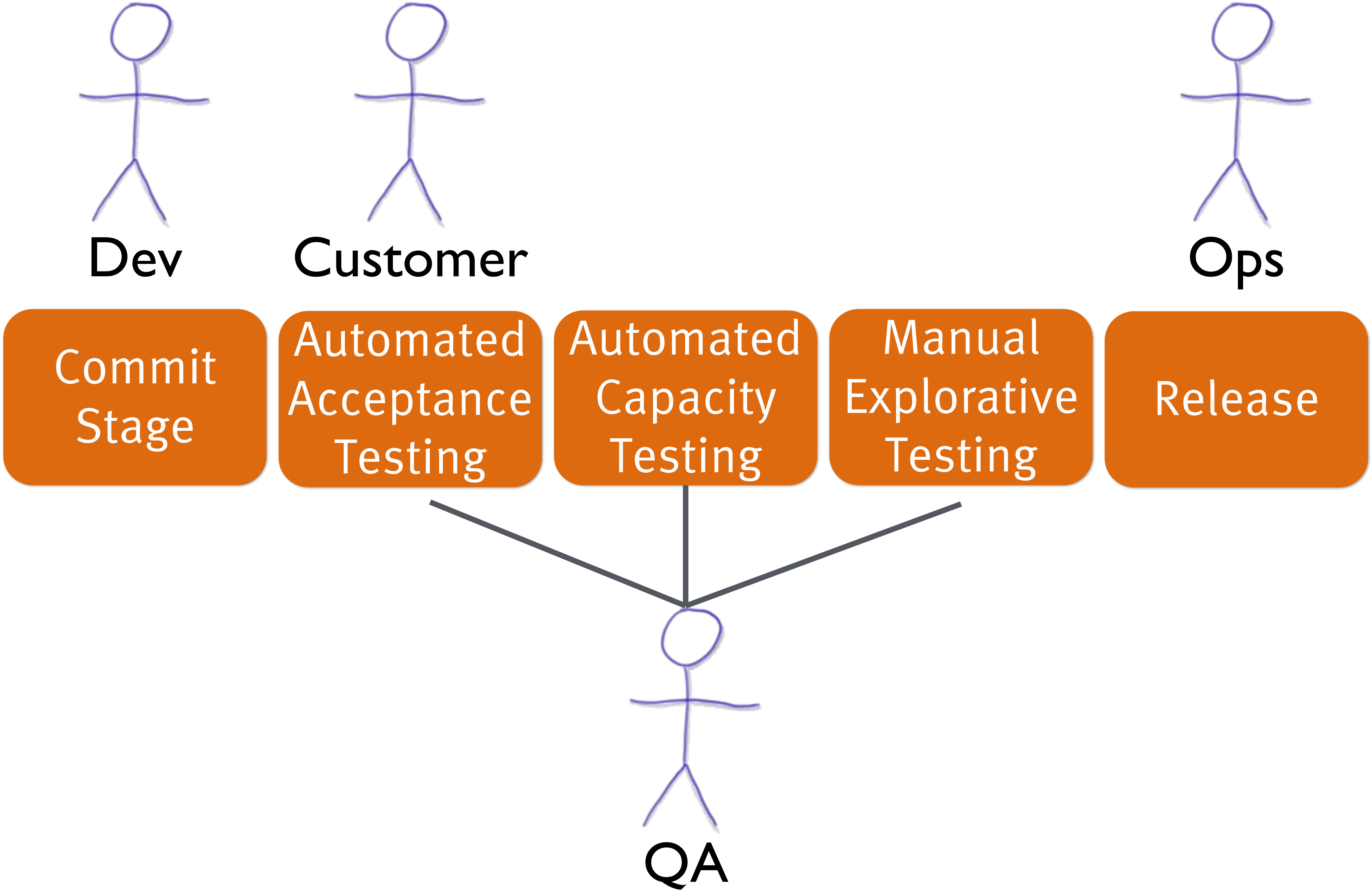
Manual Explorative Testing

Release

QA

PaaS

- Cloud Foundry, Openshift, Kubernetes
- Install a PaaS once (challenge)
- All future deployments via PaaS
- Technology to solve the social DevOps issue
- ...but is there any disadvantage?


Dev

Customer

Ops

Commit Stage

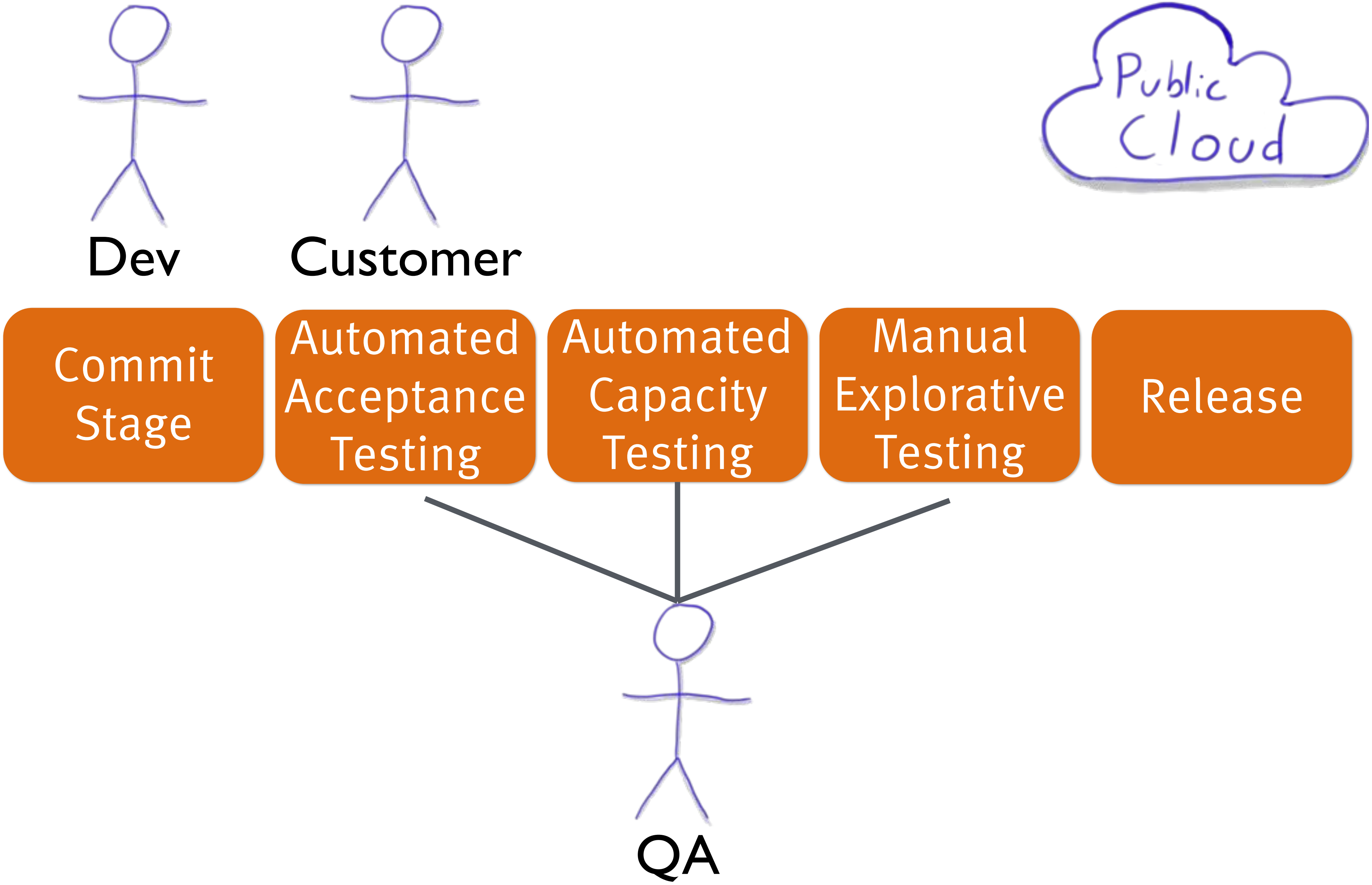
Automated Acceptance Testing

Automated Capacity Testing

Manual Explorative Testing

Release

QA



Dev

Customer

Commit Stage

Automated Acceptance Testing

Automated Capacity Testing

Manual Explorative Testing

Release

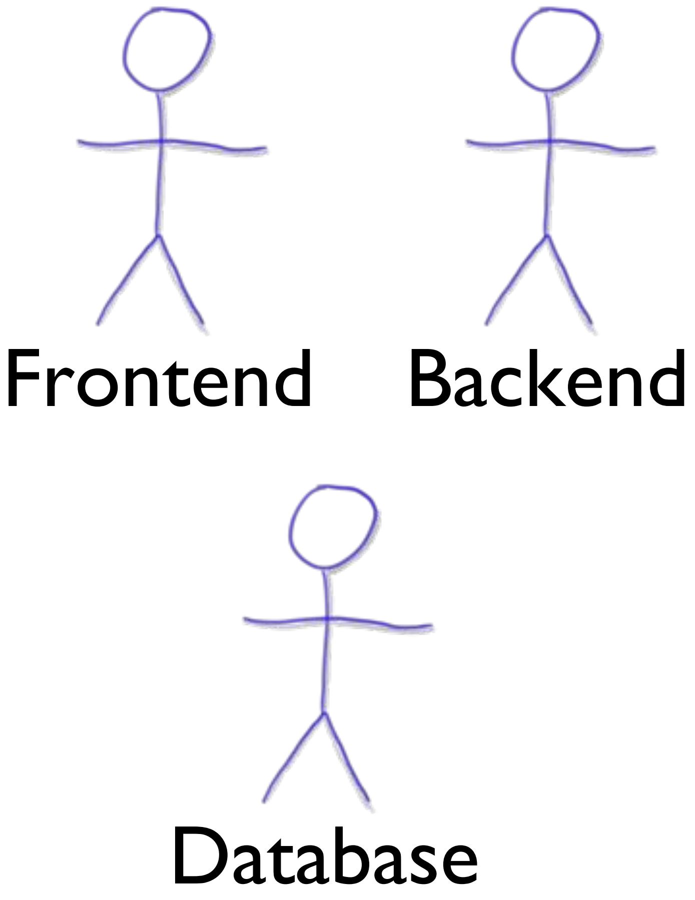
QA

Public Cloud

- Two minutes for a database instead of one month
- Many predefined offerings for Big Data, messaging…
- …but off premise
- A problem or a strategy to keep your job?

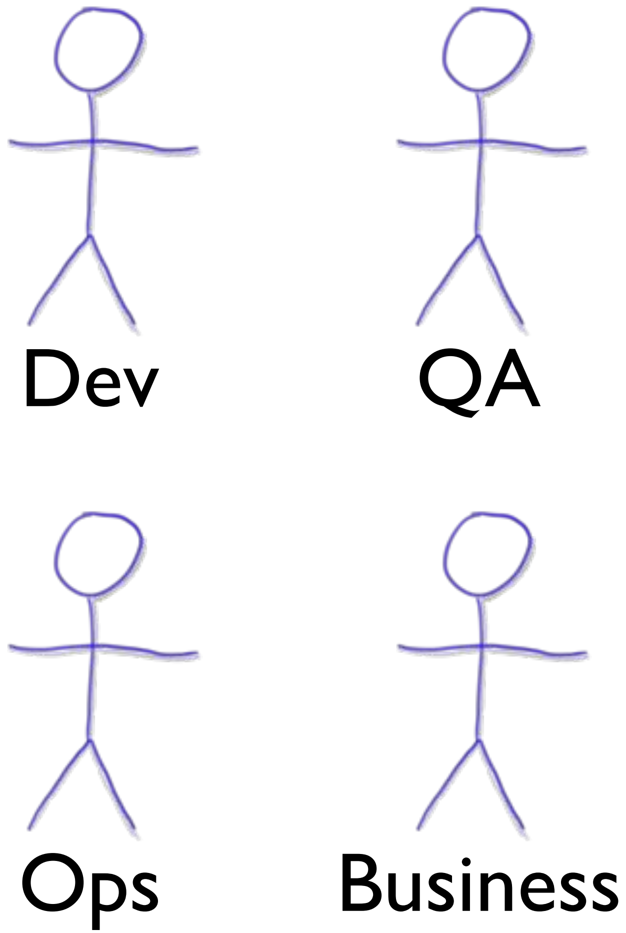
Cross-functional Team

- Was: teams with broad skill set
- i.e. frontend, backend, database
- Benefits agility:
  Can work on meaningful business features



More Cross-functional Team

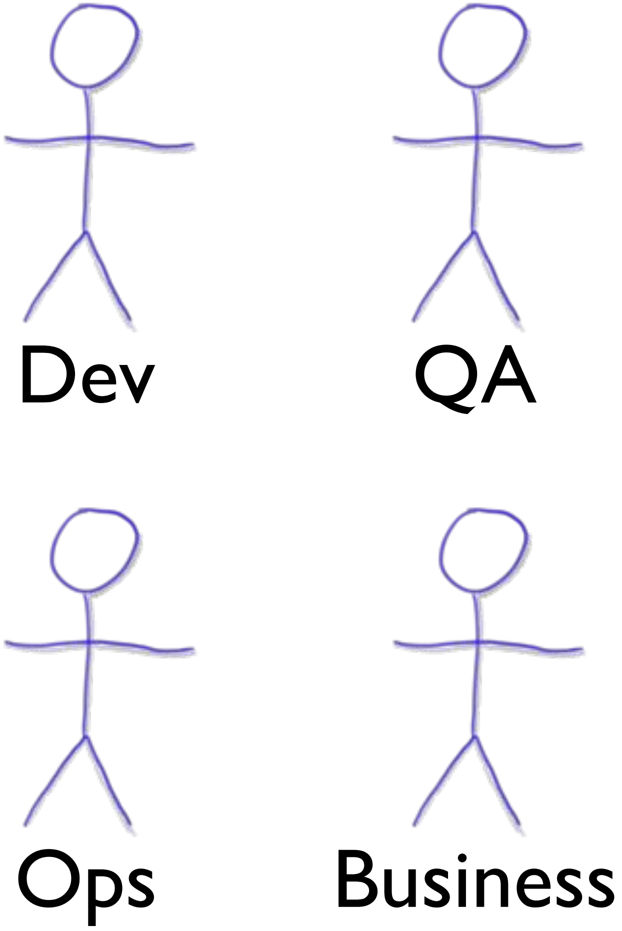
- Include QA, Ops
- ...even business
- Might build guilds to foster knowledge exchange
- Spotify



More Cross-functional Team

- Can be led by business goals
- Can enable self organization

- Huge organizational shift
- What happened to managers???
- Management buy-in?



Frontier: Management Buy-in

Just like Agility

Agility in the Nineties

- Grassroots movement
- The future of development!
- Teams want to do it.

- Management: Na, how can you delivery software without a huge sophisticated plan?

Agility Now

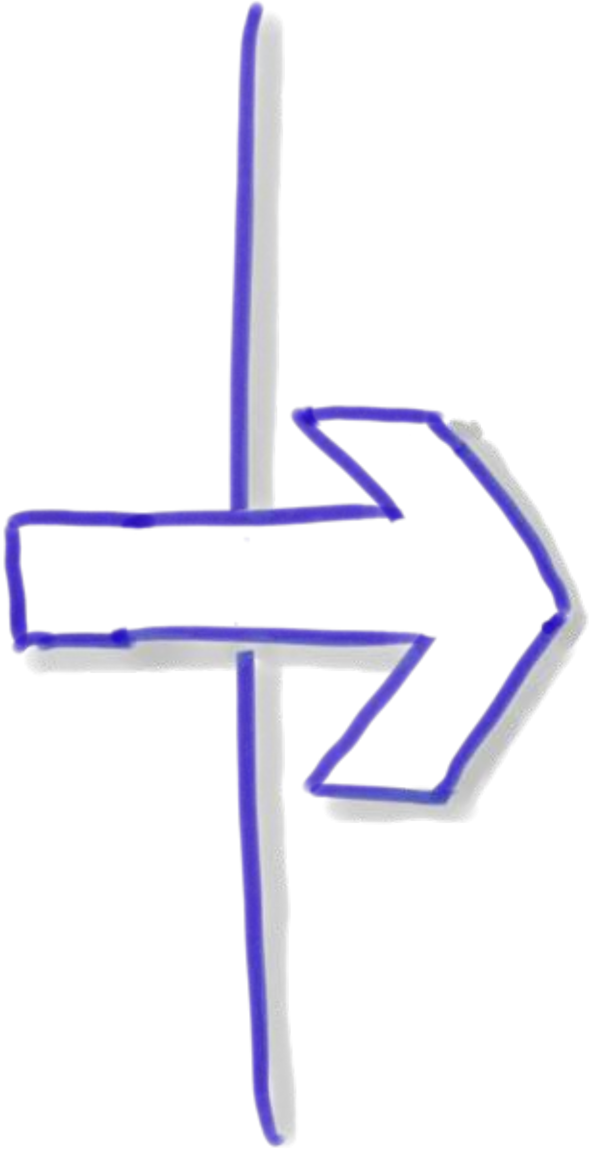
- Management: We do Scrum
- Teams skeptical or uninterested
- Business finds it hard to reap the benefits
- Still traditional product development.

Agility Now

- Need more than lip service
- …convincing

- http://blog.johanneslink.net/2011/12/02/say-goodbye-i-wont-be-back/

Extending the frontier

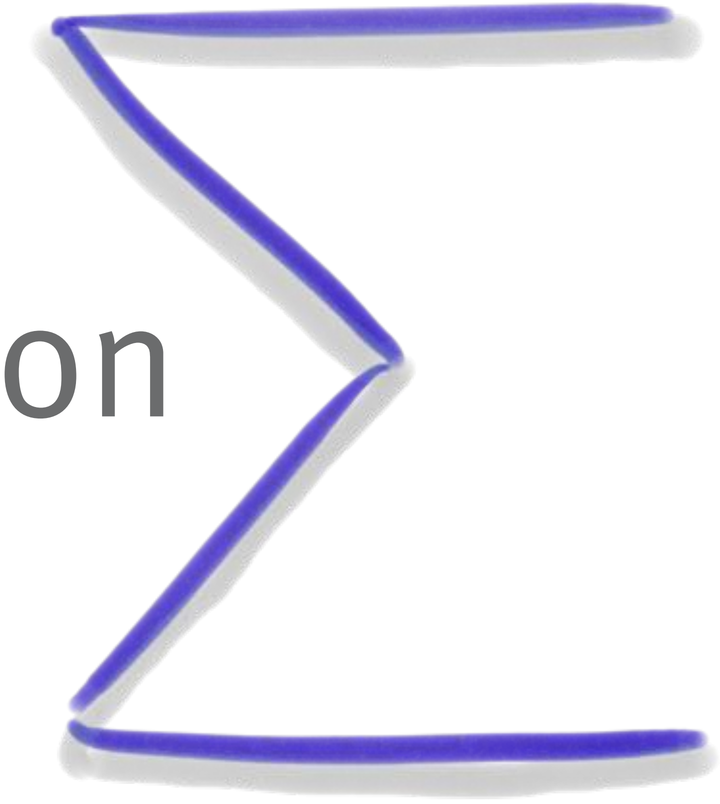


CD & Management Buy-In

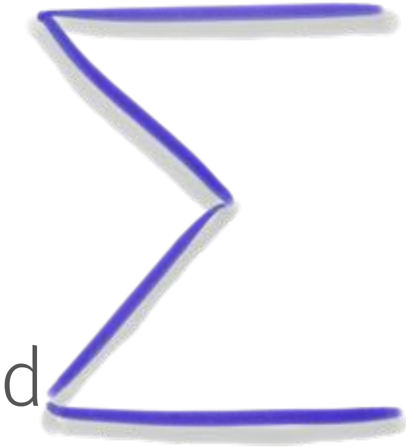
- Management buy-in won‘t solve the problems!
- It just means there will be other problems.

- Still: try to convince management.

Conclusion



Conclusion



- Technological problems mostly solved
- Microservices might support Continuous Delivery.

People Continuous Delivery Business

! ?

(Management Buy-In)

Continuous Delivery is People.

Gerald Weinberg‘s
2nd Law of Consulting:

No matter how it looks at
first,
it's always a people
problem.

EMail bedcon2017@ewolff.com to get:

Slides

+ Microservices Primer

+ Sample Microservices Book

+ Sample of Continuous Delivery Book

Powered by Amazon Lambda & Microservices