innoQ

MicroServices

meet Real World projects

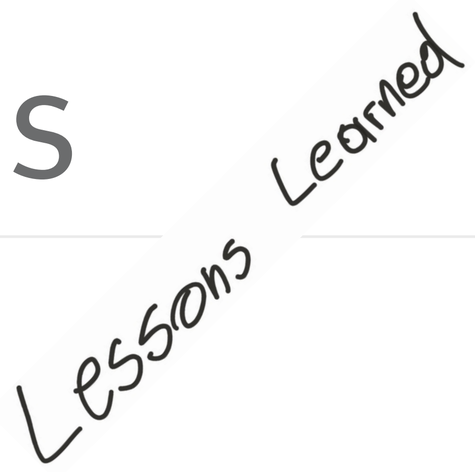
Tammo van Lessen | tammo.vanlessen@innoq.com

Alexander Heusingfeld | alexander.heusingfeld@innoq.com

innoQ

MicroServices

meet Real World projects

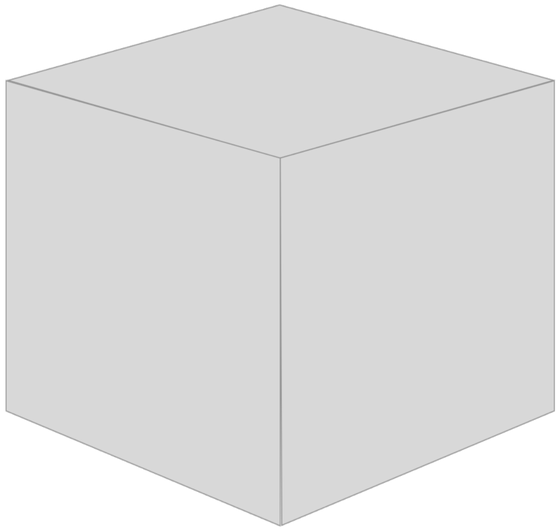


Tammo van Lessen | tammo.vanlessen@innoq.com

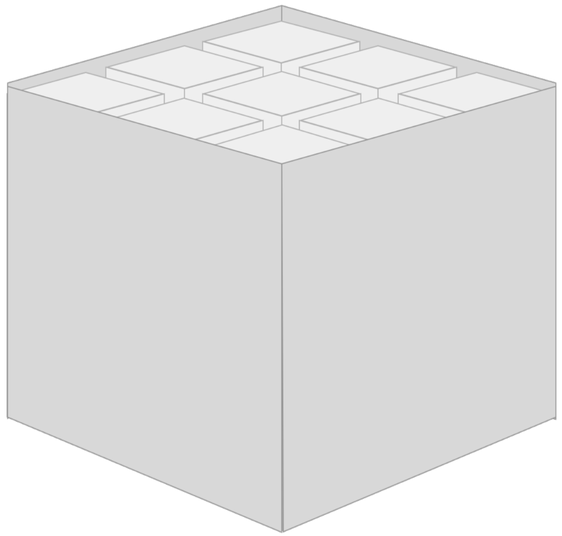
Alexander Heusingfeld | alexander.heusingfeld@innoq.com

Consultings gigs...

“We’d like to have a microservice architecture!”

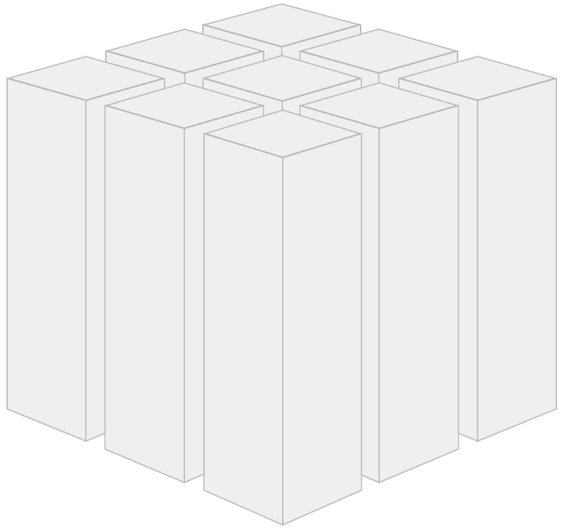


When reviewing a monolithic application ...

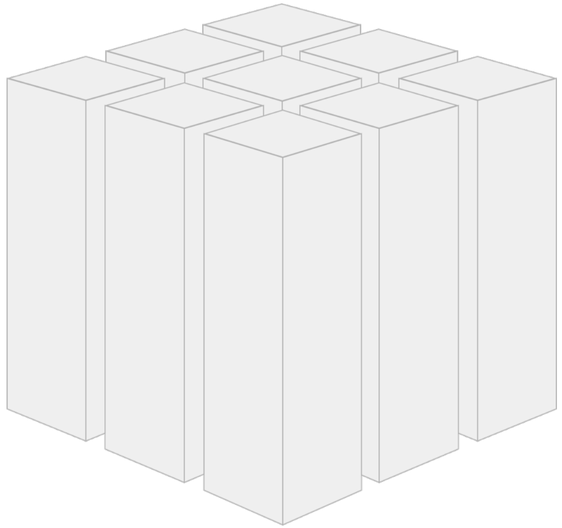


...and taking a look

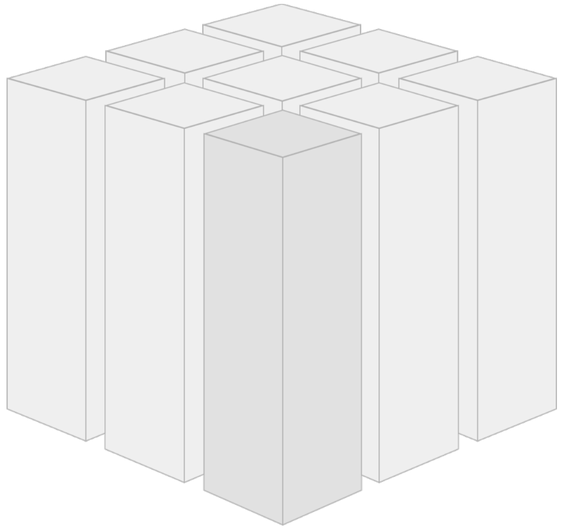
into the black box...



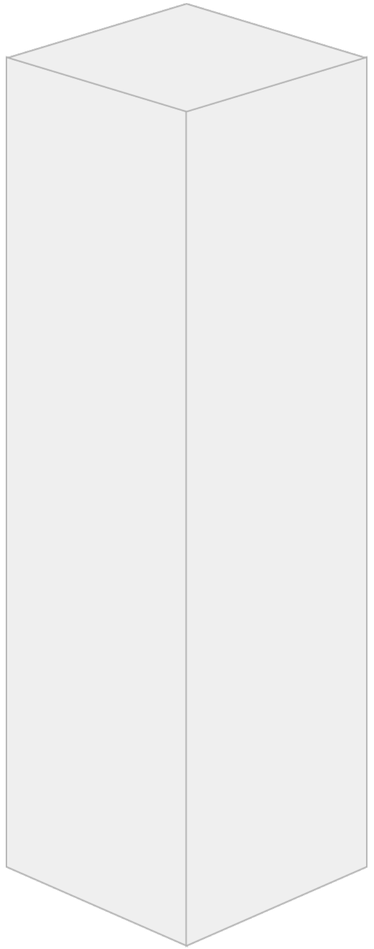
...you'll likely find it consists of multiple Bounded Contexts.



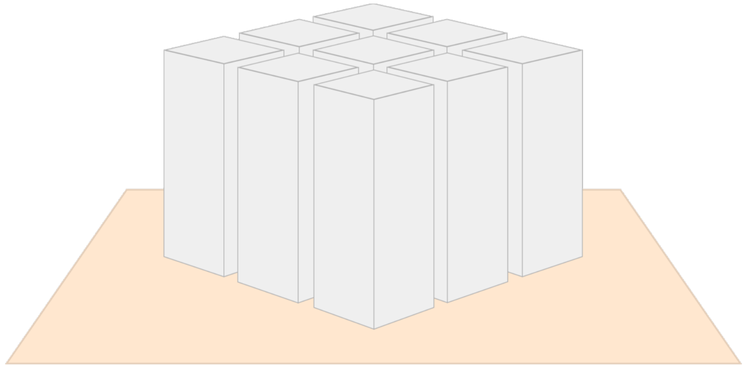
If you cut a monolithic system along its very domains ...



... and wrap every domain in
a **separate, replaceable**
web application ...



... then that application can be referred to as a **self-contained system** (SCS).



more information on

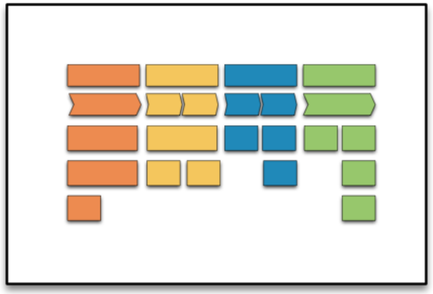
self-contained systems (SCS)

can be found at

https://www.innoq.com/de/links/self-contained-systems-infodeck/

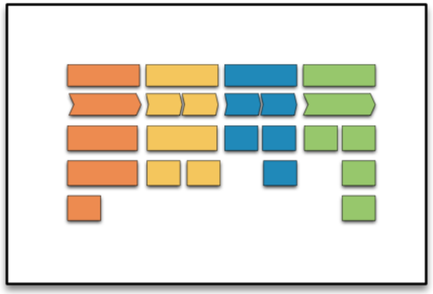
Architectural Decisions

Architectural Decisions

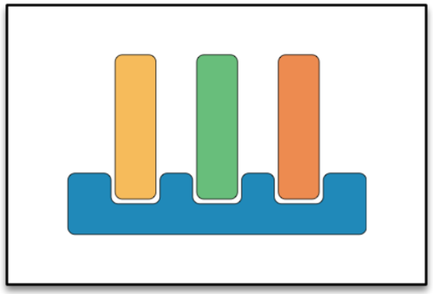


- Domain Architecture

Architectural Decisions

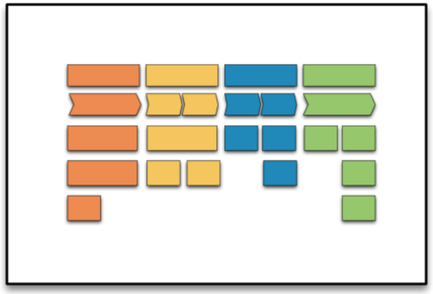


- Domain Architecture

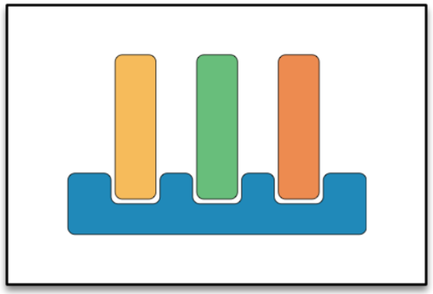


- Macro Architecture

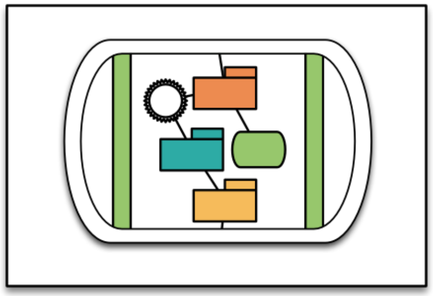
Architectural Decisions



› Domain Architecture



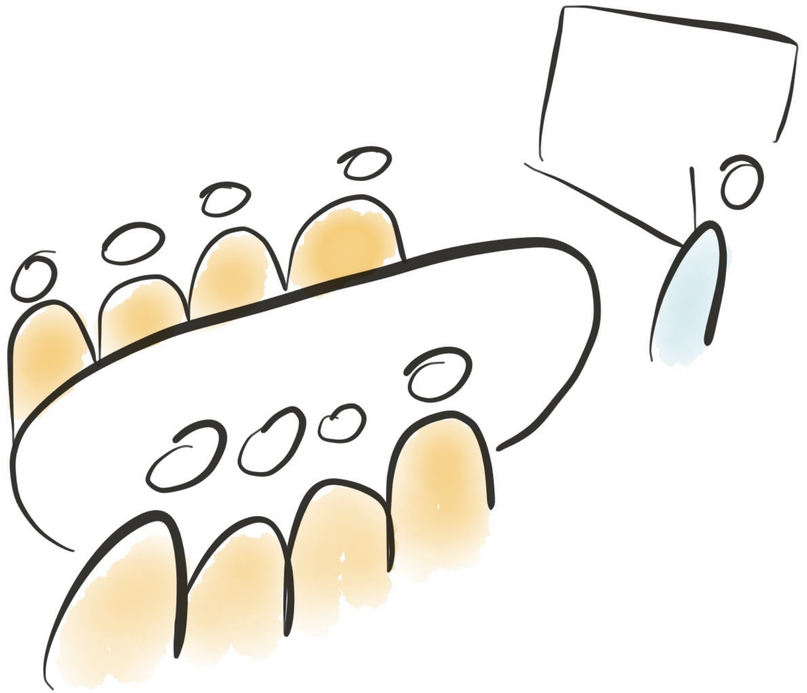
› Macro Architecture



› Micro Architecture

Isn't there more than that...

At a project meeting...



Did you think about the people

who make your architecture exist?

us vs. them

“Don’t care about this, it’s our business!”

“Alarming is our concern, don’t bother about it!”

“No need for a discussion, we always fix that during deployment.”

“That’s part of the handover to operations.”

overcome “us vs. them”

overcome “us vs. them”

- cross-functional != cross-department

overcome “us vs. them”

- cross-functional != cross-department
- have one manager to decide on a team’s targets

overcome “us vs. them”

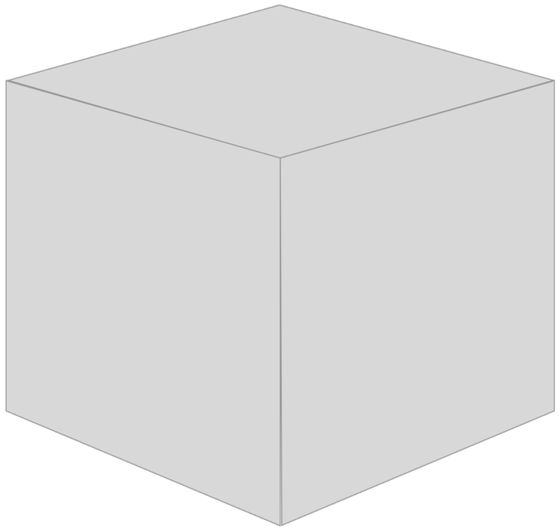
- cross-functional != cross-department
- have one manager to decide on a team’s targets
- don’t neglect team-building

overcome “us vs. them”

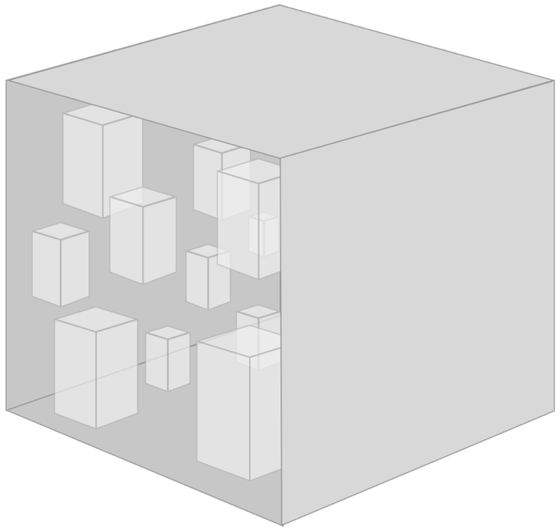
- cross-functional != cross-department
- have one manager to decide on a team’s targets
- don’t neglect team-building
- trust is not optional

well-known pros are subjective

“Operating a monolith is easier!”



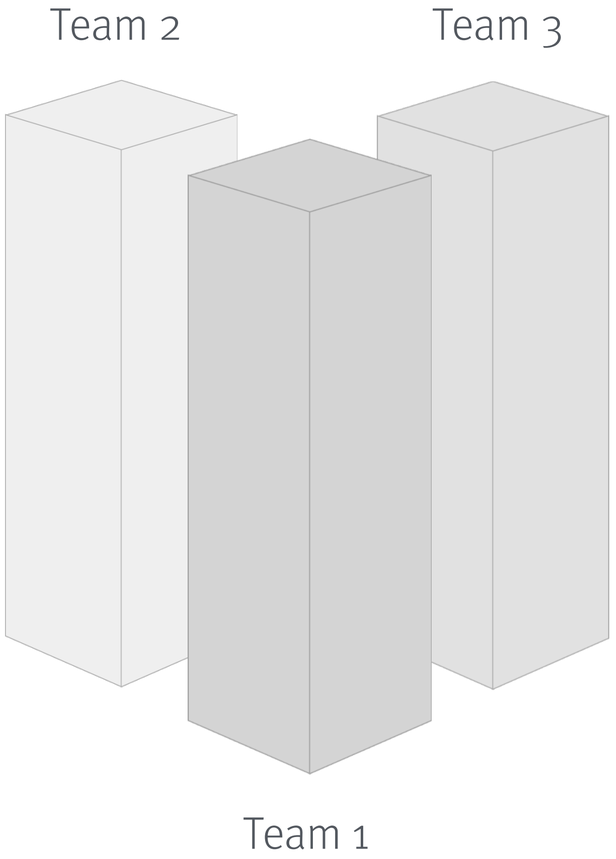
Of course it’s easier...



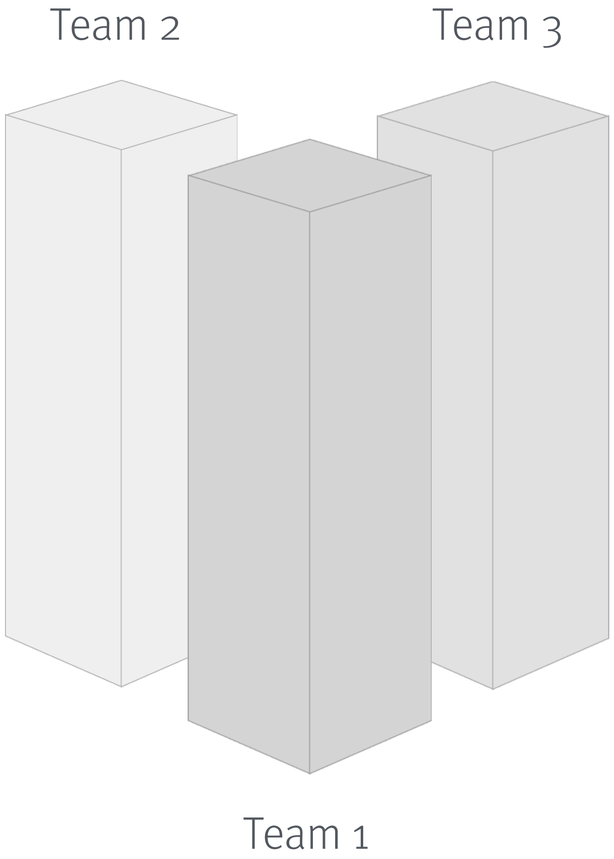
It’s always easier...

...if the complexity is on someone else’s desk.

“Separating teams duplicates work!”



The manageable, domain specific scope enables the development, operation and maintenance of an SCS by a **independent team**.



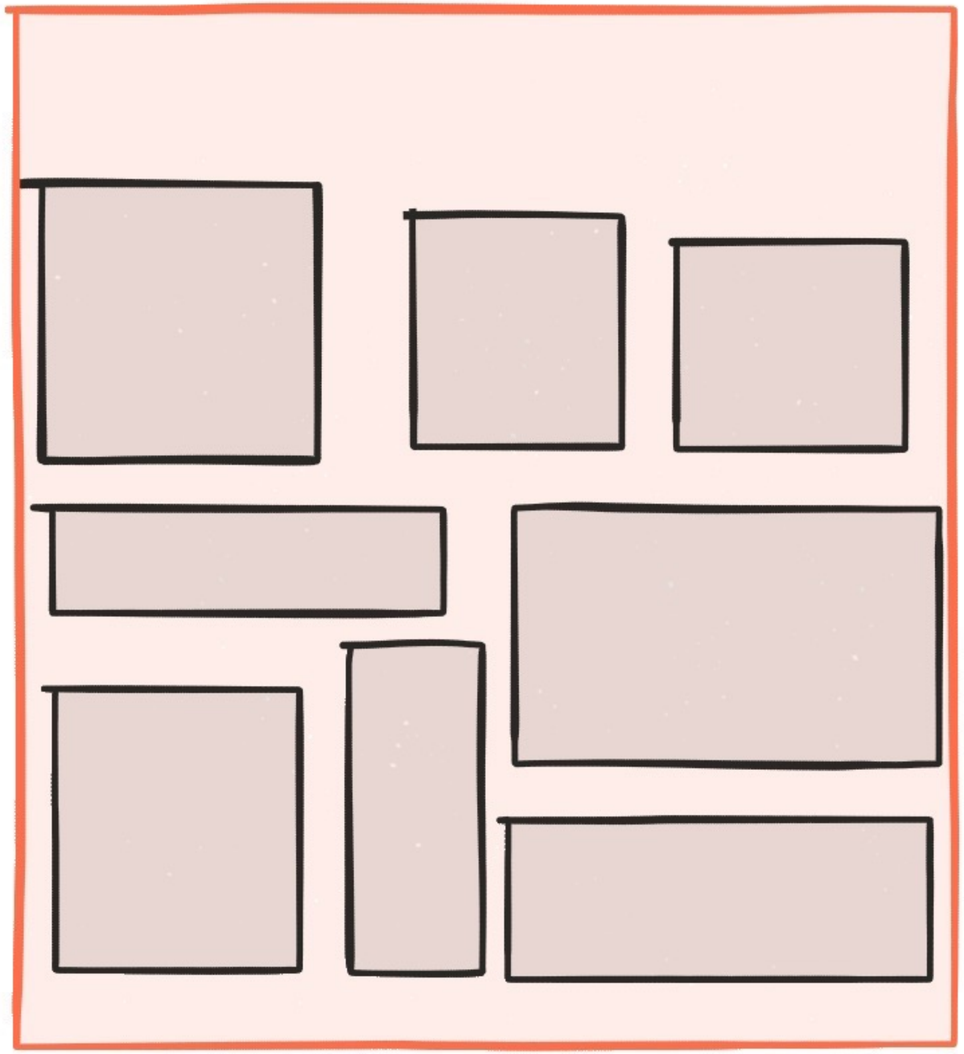
share ideas

share concepts

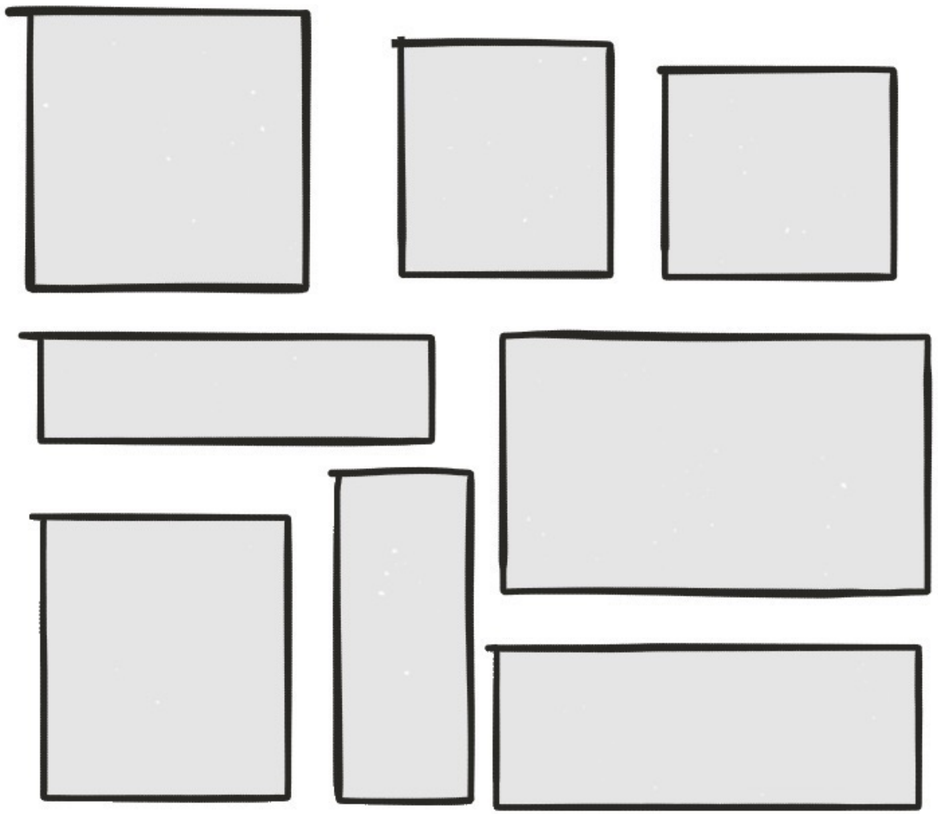
don’t share libraries

“Operational costs are increased!”

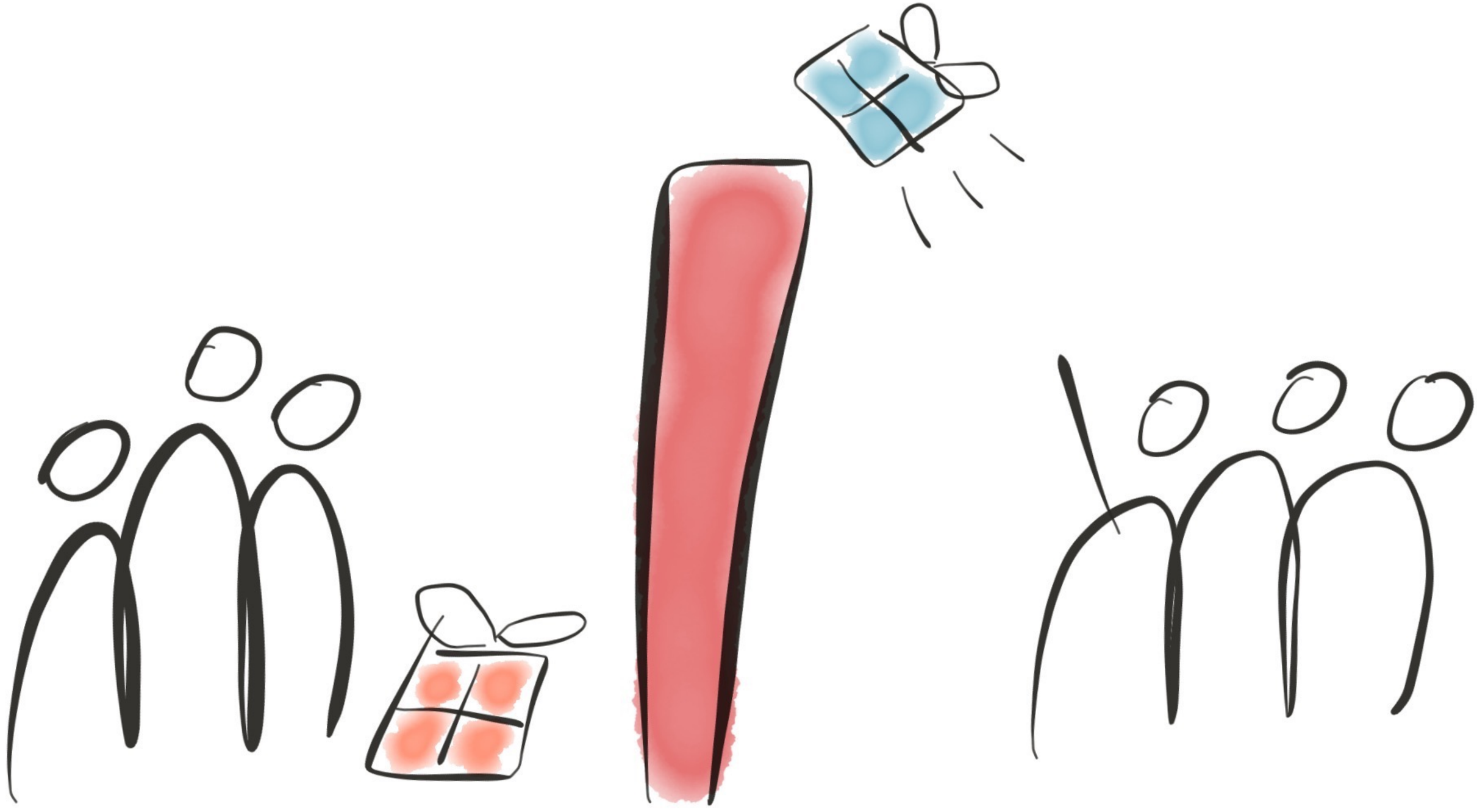
Monolith



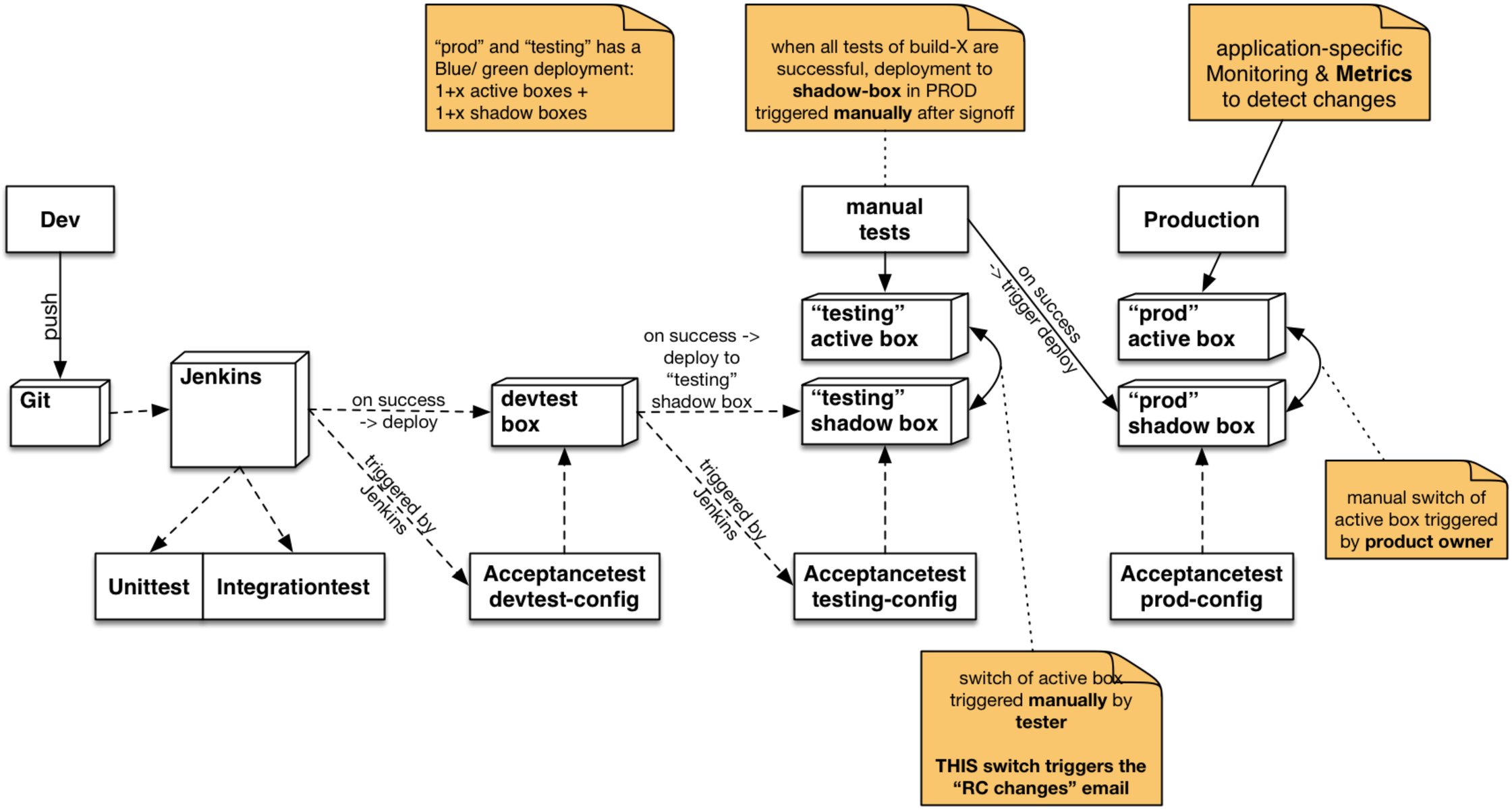
Microservices?



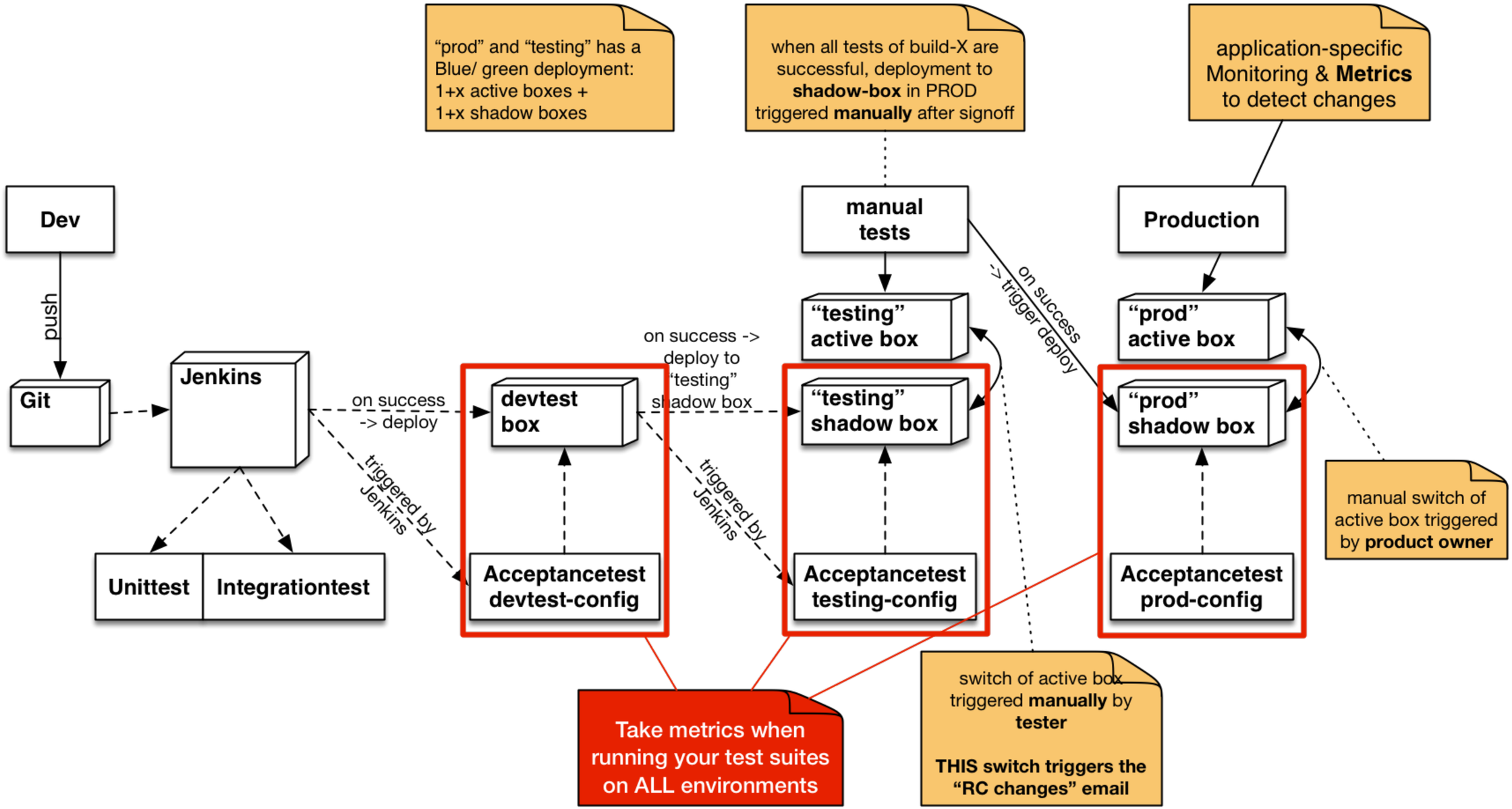
“Deployments cannot be faster,
we have an established process!”



Sample of a deployment-pipeline



Sample of a deployment-pipeline



What this taught us

What this taught us

- enable fast feedback for your team

What this taught us

- enable fast feedback for your team
- automate what’s next to you first

What this taught us

- enable fast feedback for your team
- automate what’s next to you first
- do your homework before you teach others

What this taught us

- enable fast feedback for your team
- automate what's next to you first
- do your homework before you teach others
- other people will notice the benefits

What this taught us

- enable fast feedback for your team
- automate what's next to you first
- do your homework before you teach others
- other people will notice the benefits
- complex processes can be adopted, divide them and take one step at a time

“pets vs. cattle”

https://www.flickr.com/photos/cornelii/531691572

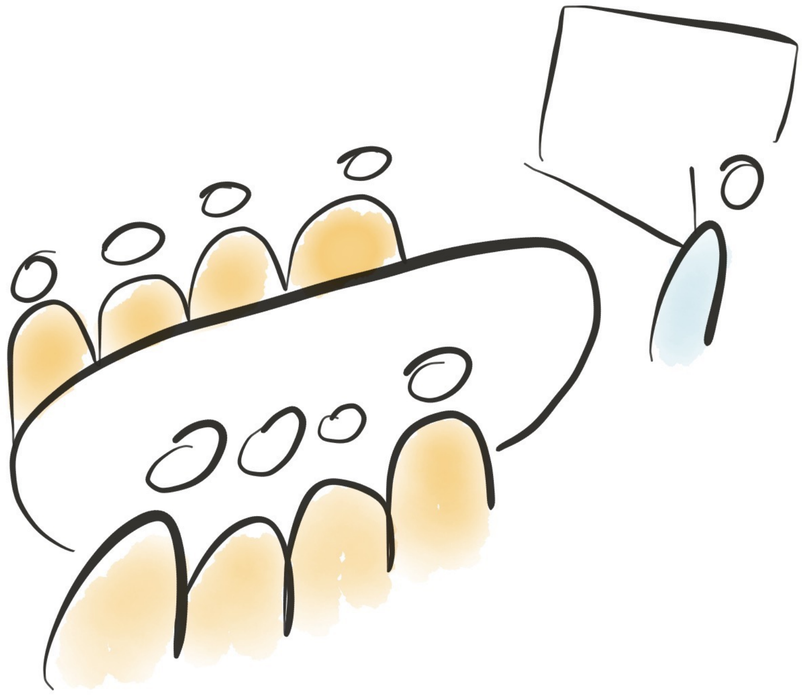
https://www.flickr.com/photos/cornelii/531691572

https://www.flickr.com/photos/cornelii/531691572

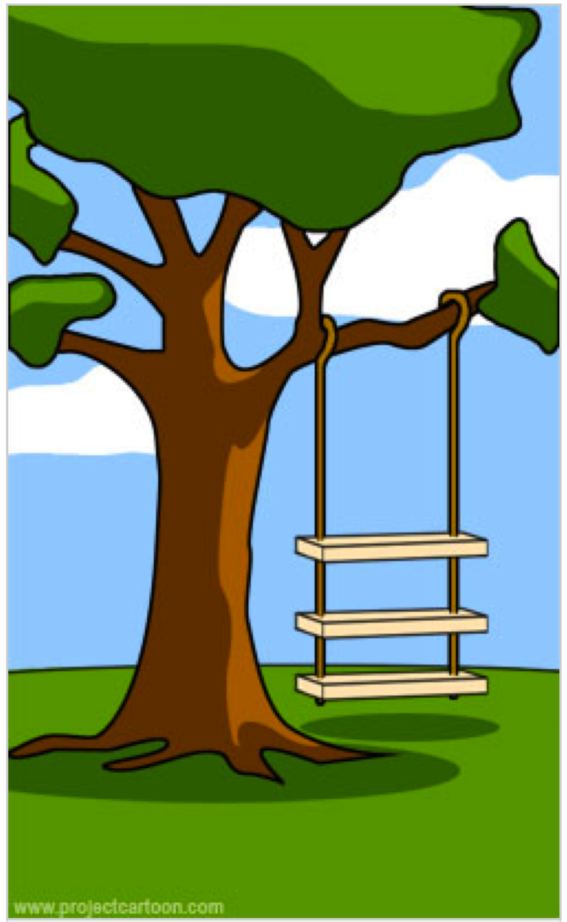
https://www.flickr.com/photos/cornelii/531691572

summarized:
change perspectives!

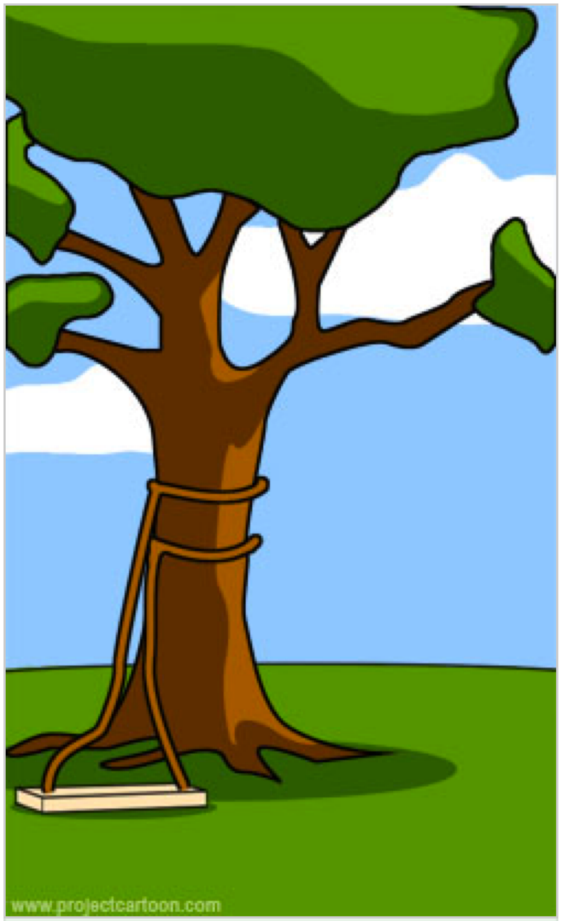
A company which
embraced and evolved



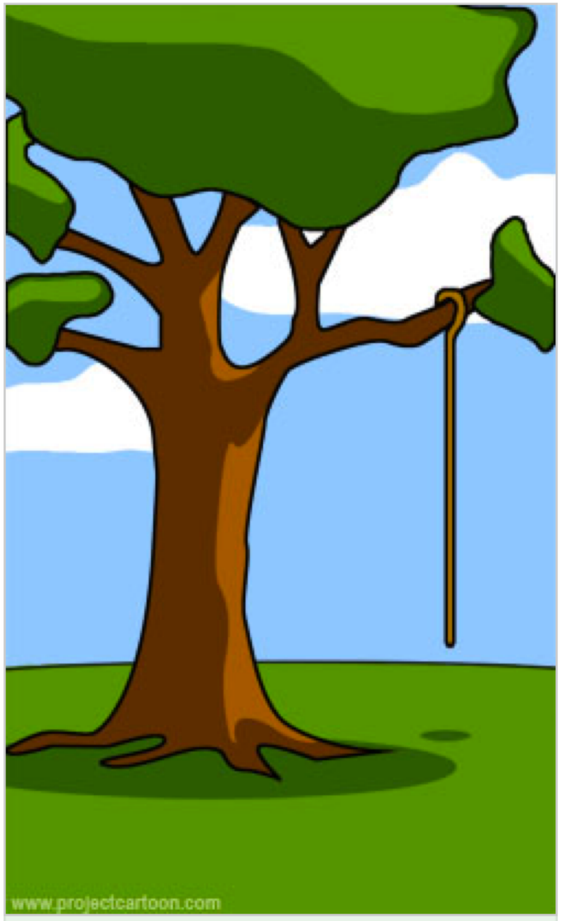
http://projectcartoon.com/cartoon/1



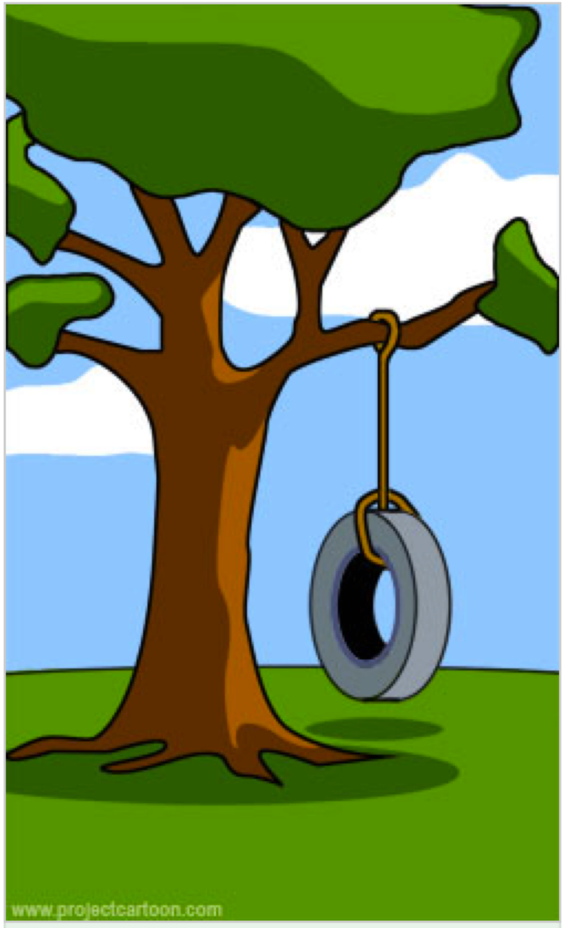
How the customer explained it



How the programmer wrote it



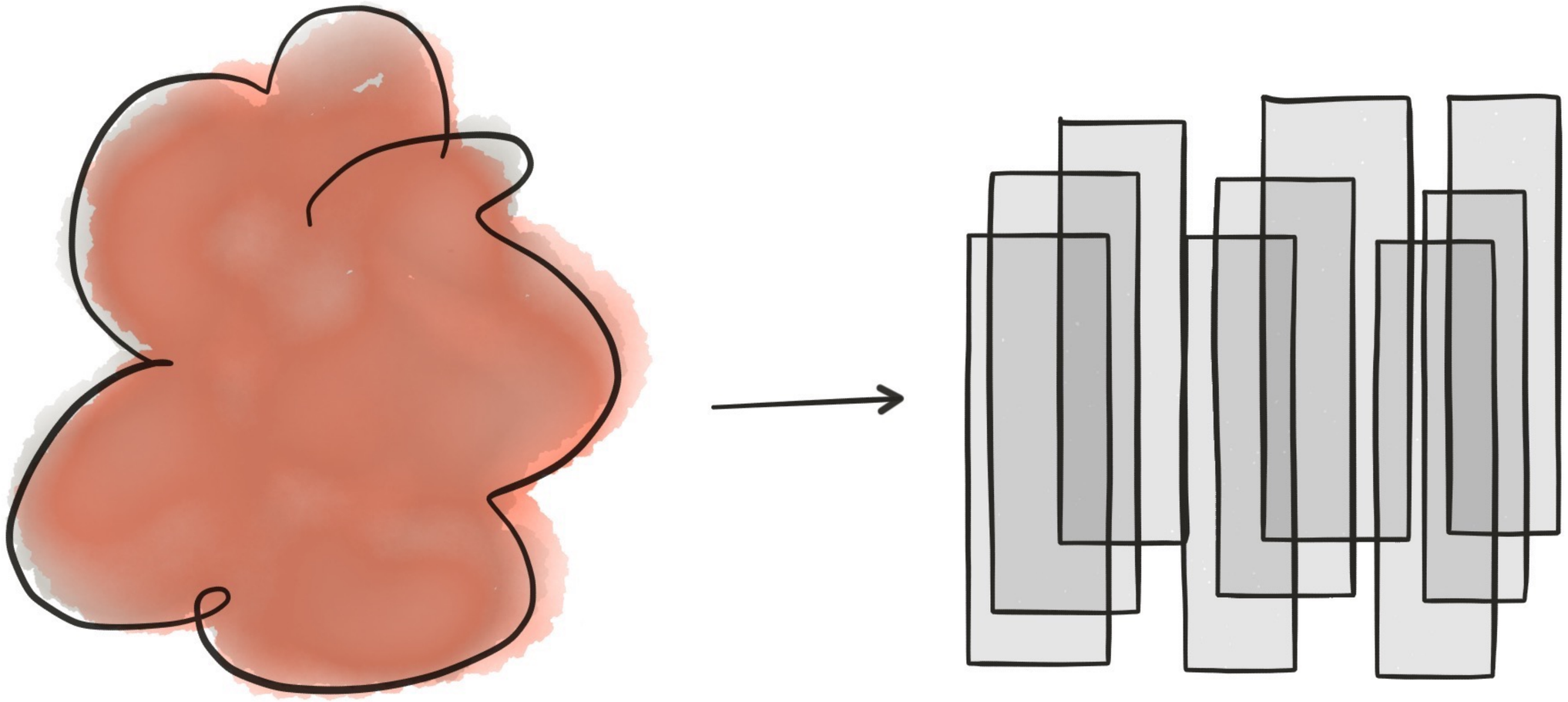
What operations installed



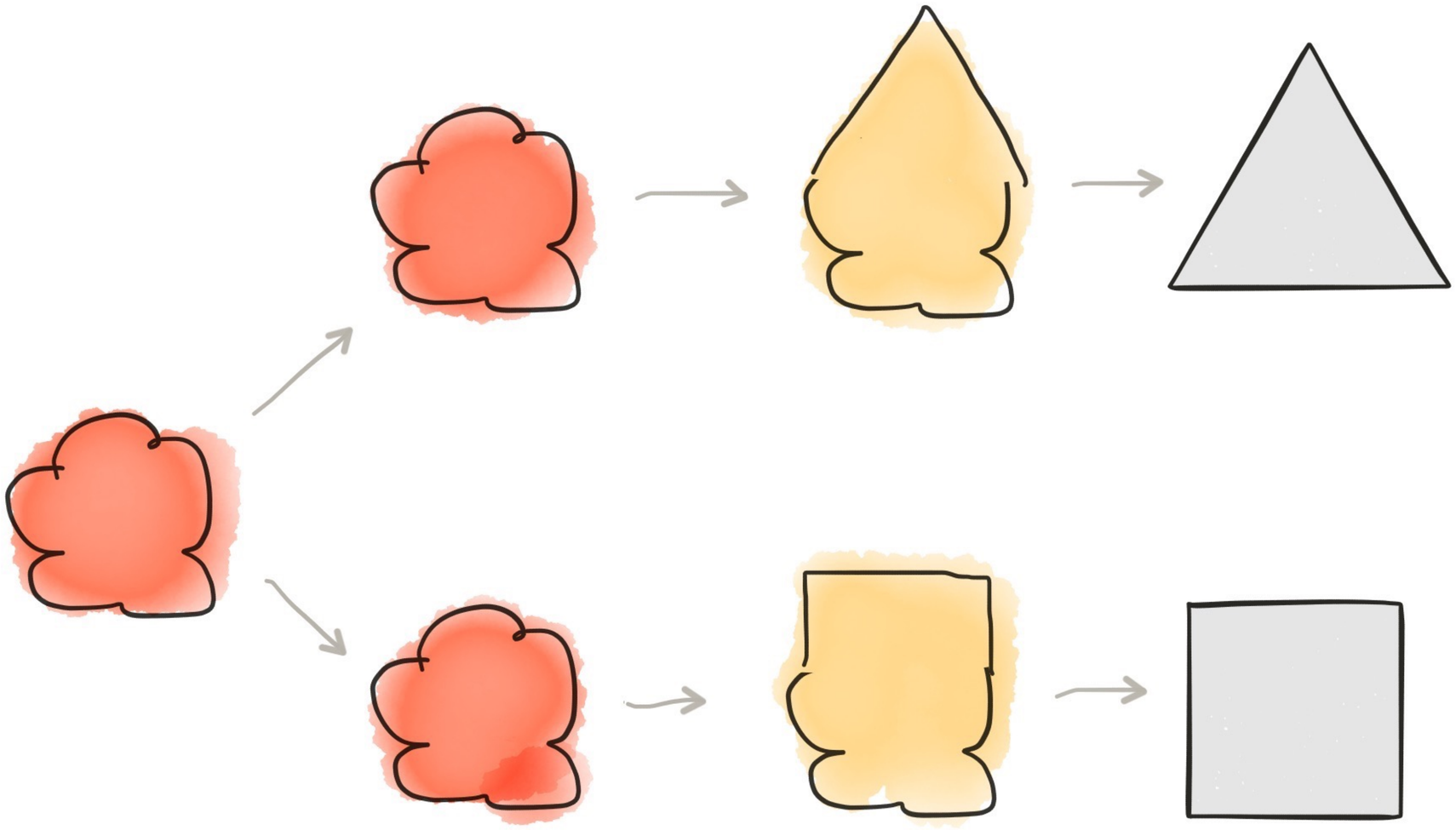
What the customer really needed

Modernisation Strategies

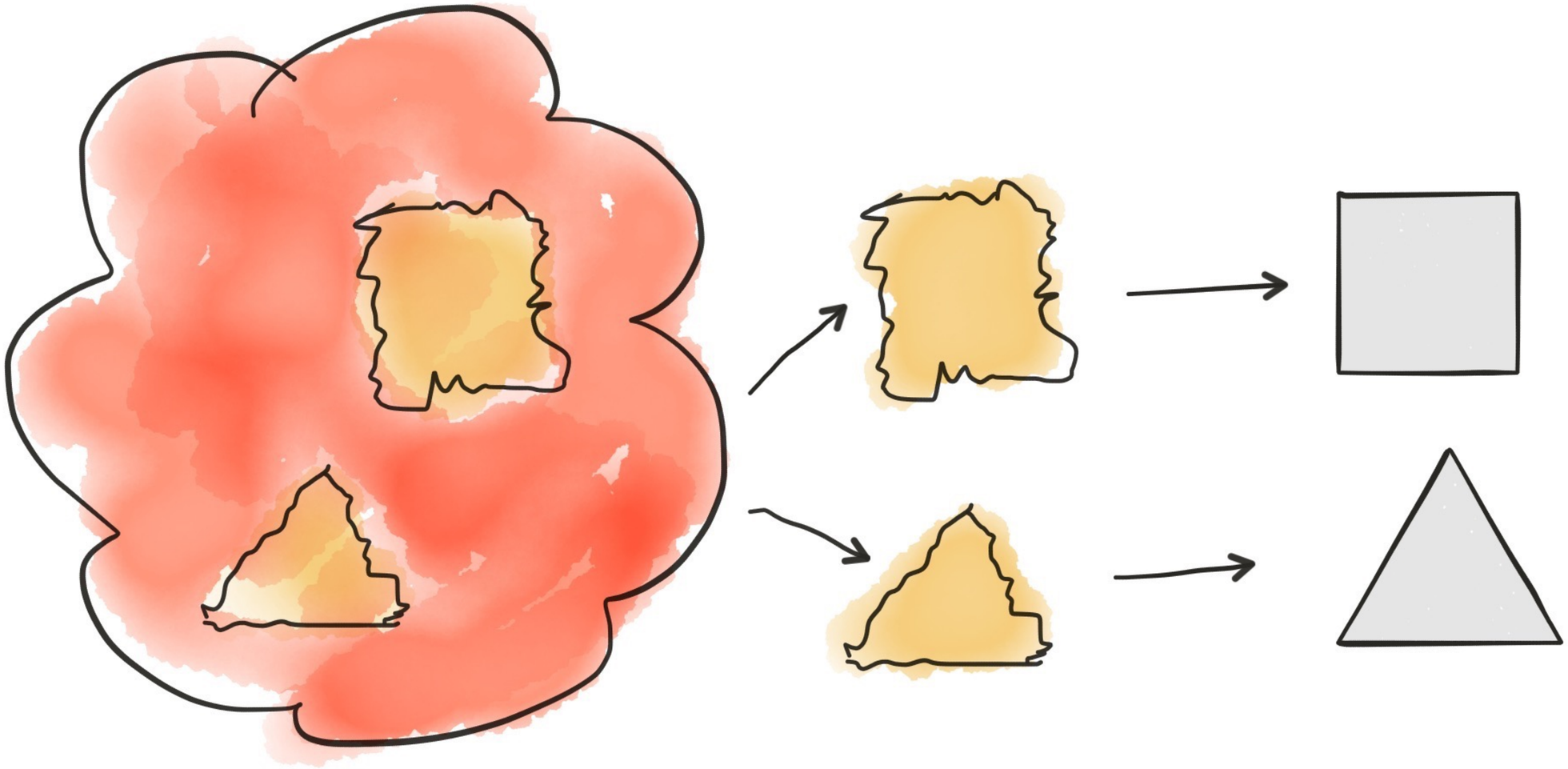
Big Bang



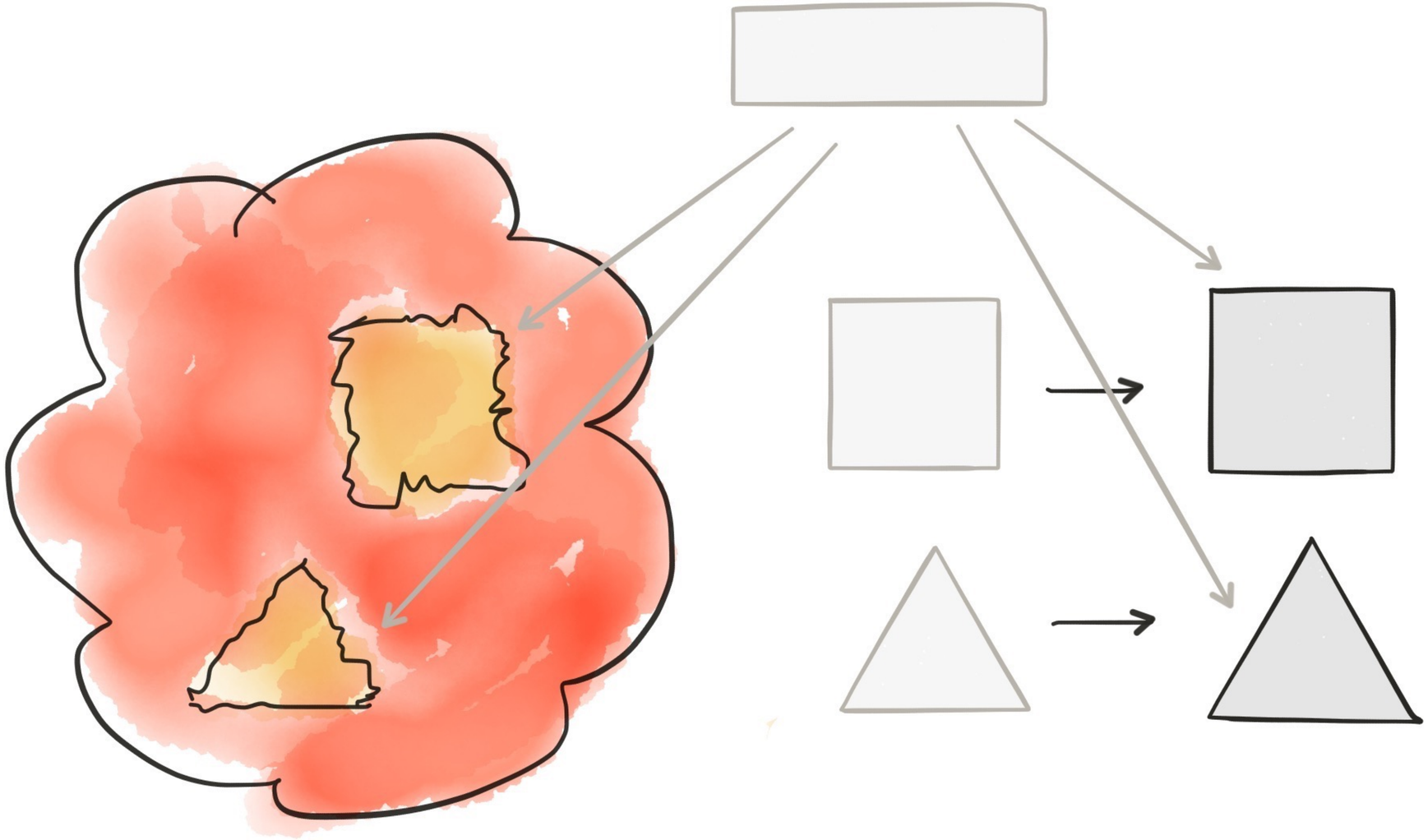
Change via Copy



Change via Extraction



Strangulate Bad Parts



more information on

software modernisation

can be found at

http://aim42.org/

conclusion

Conway's Law

Organization → Architecture

> **"Organizations which design systems are constrained to produce systems which are copies of the communication structures of these organizations." – M.E. Conway**

Summary

Summary

- distributed systems are hard - **organizational impact**, too

Summary

- distributed systems are hard - **organizational impact**, too
- don't forget: there's always at least one **other perspective**

Summary

- distributed systems are hard - **organizational impact**, too
- don't forget: there's always at least one **other perspective**
- **Don’t overwhelm people**, change one thing at a time

Summary

- distributed systems are hard - **organizational impact**, too
- don't forget: there's always at least one **other perspective**
- **Don’t overwhelm people**, change one thing at a time
- not everyone who **wants** microservices is immediately **capable** to establish them

Thank you!

Questions?

Comments?

Tammo van Lessen | @taval

tammo.vanlessen@innoq.com

Alexander Heusingfeld | @goldstift

alexander.heusingfeld@innoq.com

https://www.innoq.com/en/talks/2015/10/javaone-2015-microservices-projects-selfcontained-systems/

innoQ

www.innoq.com

innoQ Deutschland GmbH

Krischerstr. 100
40789 Monheim am Rhein
Germany
Phone: +49 2173 3366-0

Ohlauer Straße 43
10999 Berlin
Germany

Ludwigstraße 180 E
D-63067 Offenbach
Germany

Kreuzstr. 16
D-80331 München
Germany

innoQ Schweiz GmbH

Gewerbestr. 11
CH-6330 Cham
Switzerland
Phone: +41 41 743 0116

#javaone #microservices