Misconceptions about Continuous Delivery

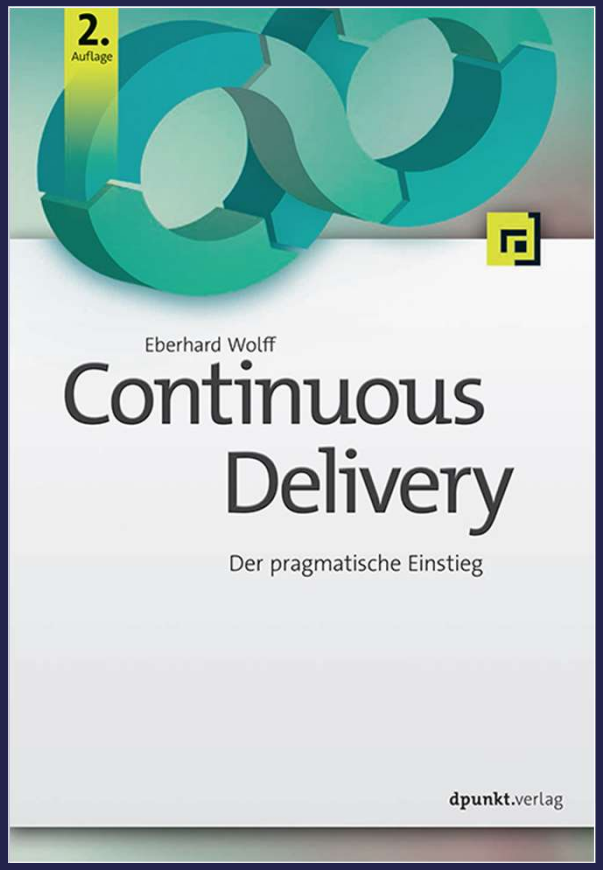
INNOQ

EBERHARD WOLFF

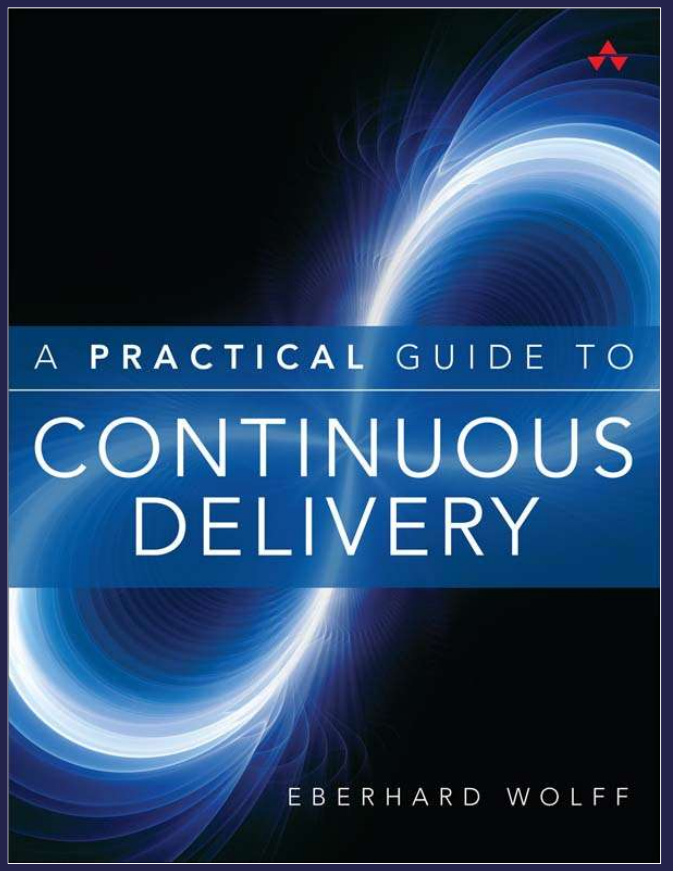
Fellow at INNOQ Deutschland GmbH

@ewolff

www.ewolff.com

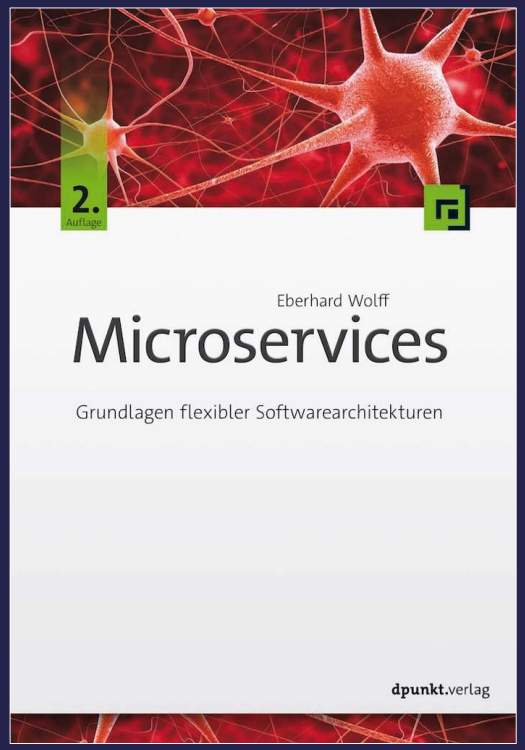


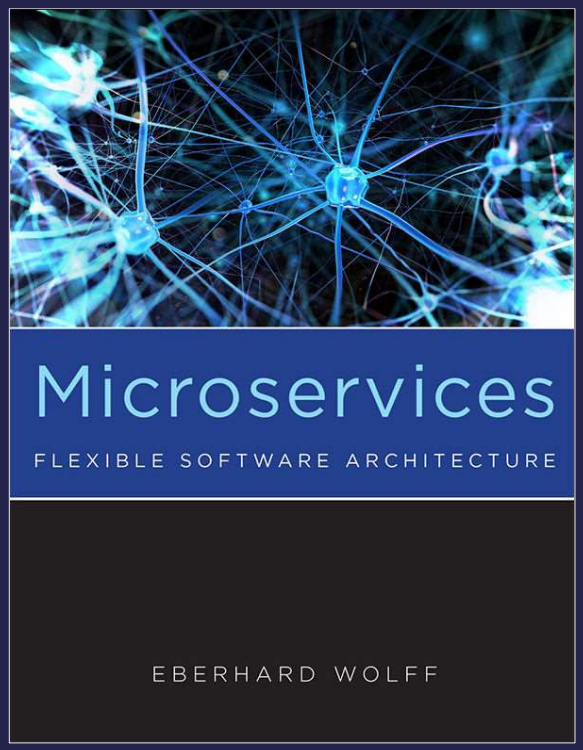
www.continuous-delivery-buch.de

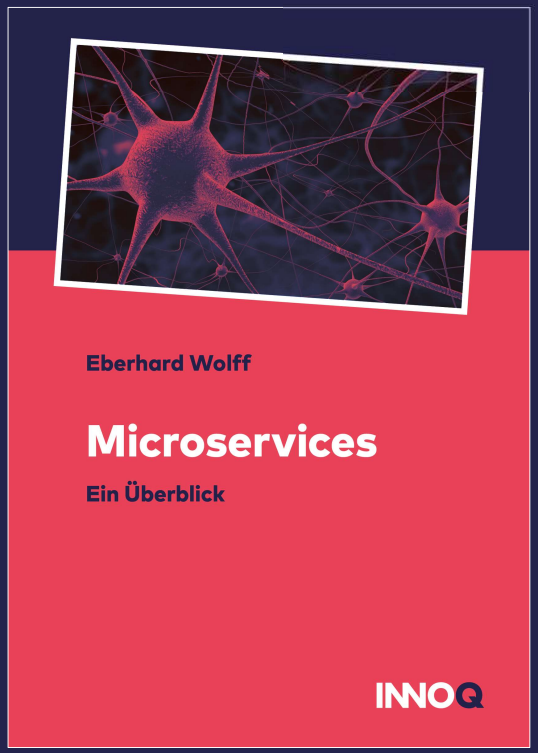


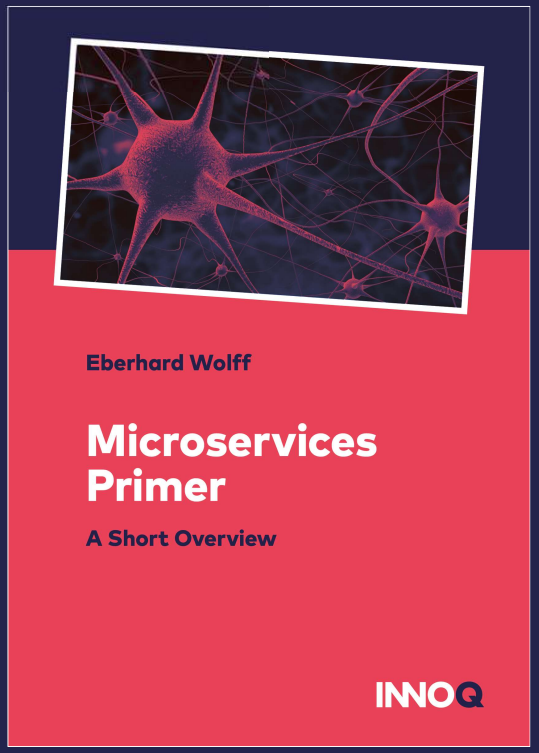
www.continuous-delivery-buch.de

FREE









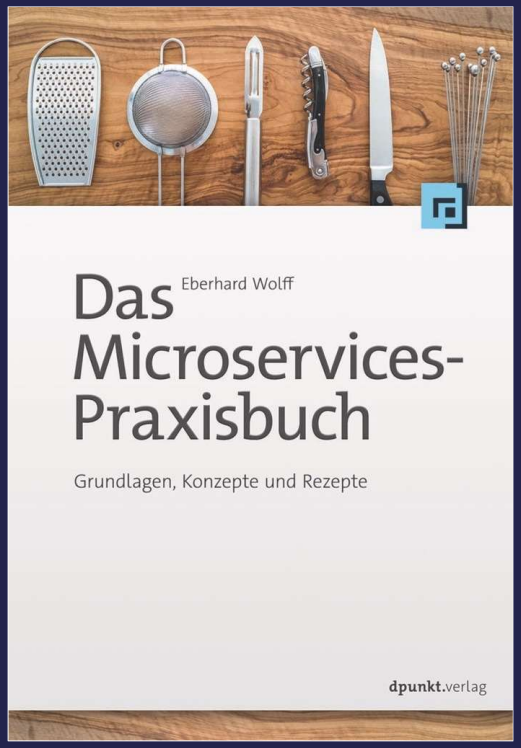
microservices-buch.de

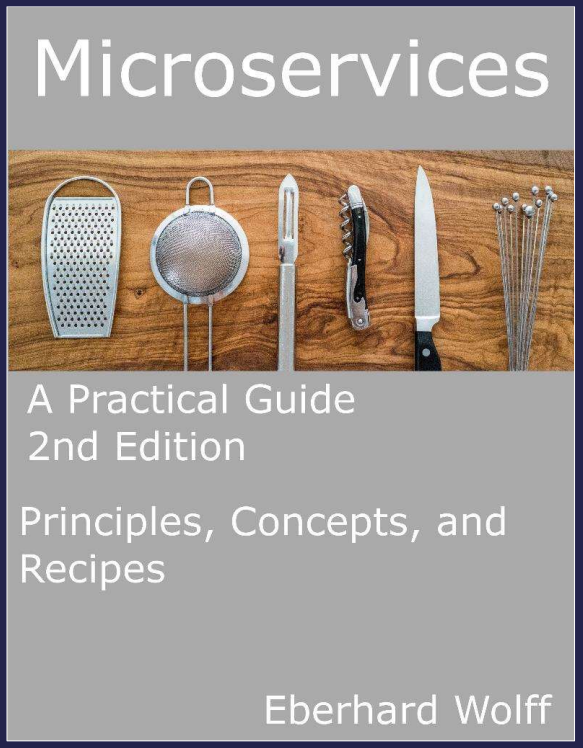
microservices-book.com

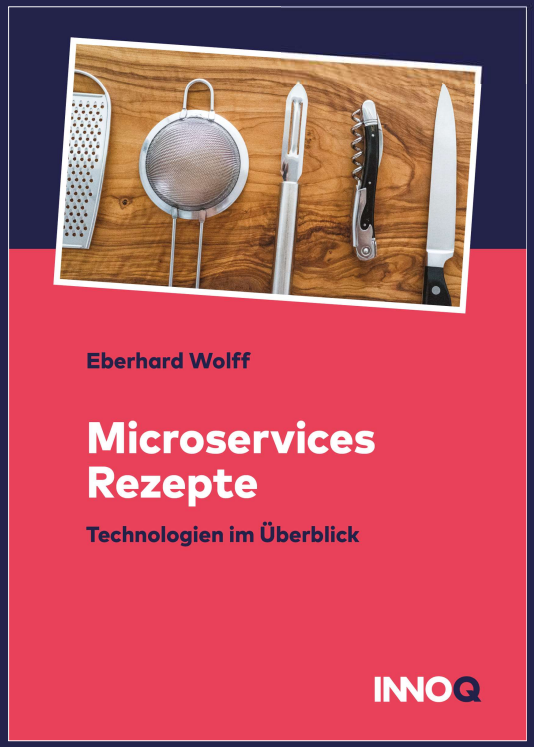
microservices-buch.de/ueberblick.html

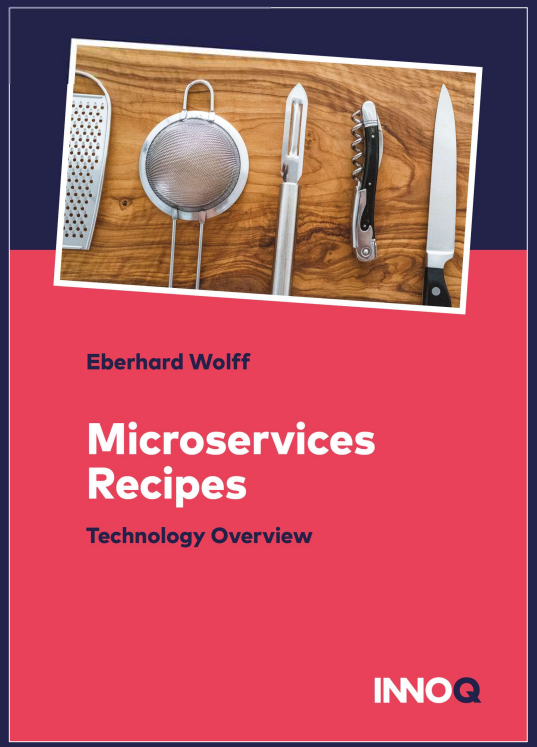
microservices-book.com/primer.html

FREE









microservices-praxisbuch.de

practical-microservices.com

microservices-praxisbuch.de/rezepte.html

practical-microservices.com/recipes.html

FREE

leanpub.com/service-mesh-primer/

Continuous Delivery Pipeline

Why This Talk?

- Continuous Delivery: since 2011
- Still not well-understood
- Agile has the same issues

- Better understanding

  ...actually reach objectives

  ...understand how CD helps

Misconception
Release = New Features

Release = New Features

- Releasing software is a risk.
- Only sensible if user has an advantage.

  i.e. new features

Features = Release is a Risk

- Features have a deadline
- Sometimes deadlines are hard

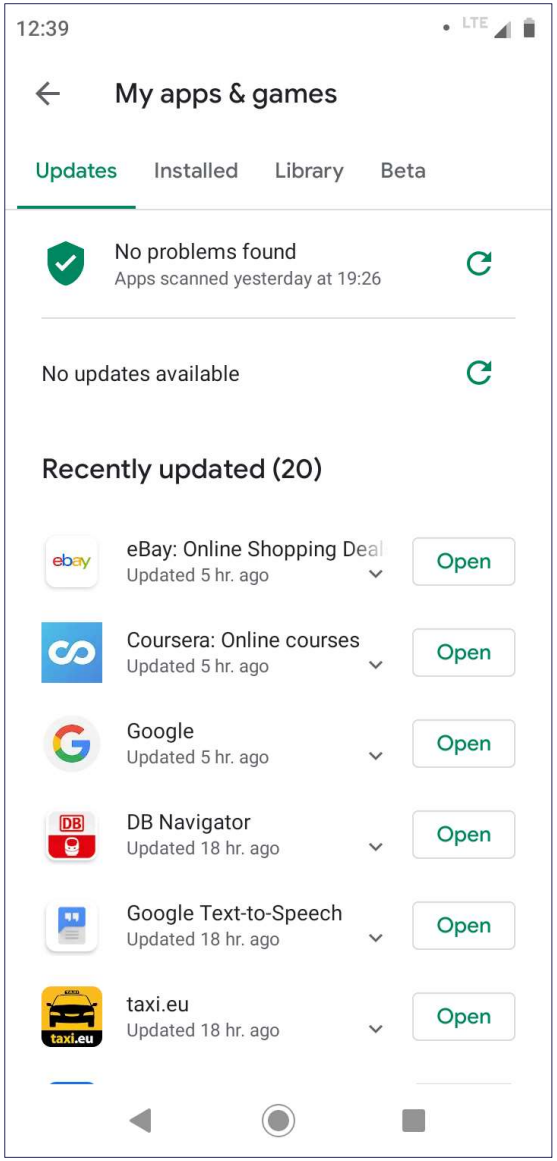
  E.g. regulations

Features = Release is a Risk

- Should release much earlier than deadline
- Try in production
- Feature Toggles to disable features for majority of users.

This Morning on My Phone...

- Not sure when I noticed a new feature the last time
- Not sure when I noticed a bug fix the last time
- Same for Windows
- Same for applications on my laptop



Why Release?

- Features seem not too important
- Are there other reasons?

Other Reasons for Releases....

- Bug fixes
- Security fixes
- New libraries with bug or security fixes.

- Continuous Delivery should lead to more security and less bugs...

Release = New Features


Release = New Features

Decouple Releases and Feature.
Other reasons for Releases.

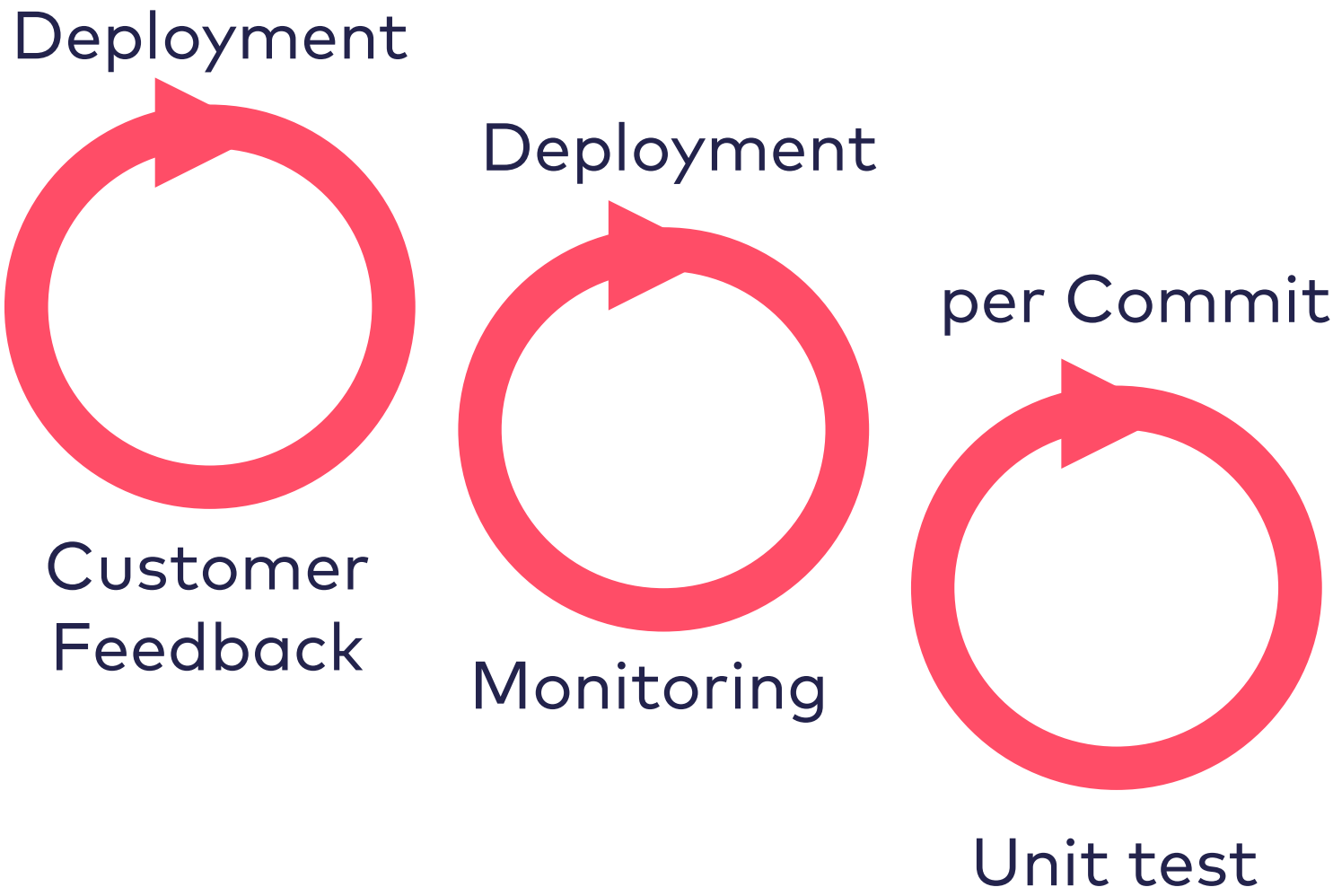
Misconception
Continuous Delivery = Time to Market

Feedback Loops

- Feedback shows whether you are on the right track
- Probably more important than big plan up front

  ...in particular in complex scenarios

Feedback Loops



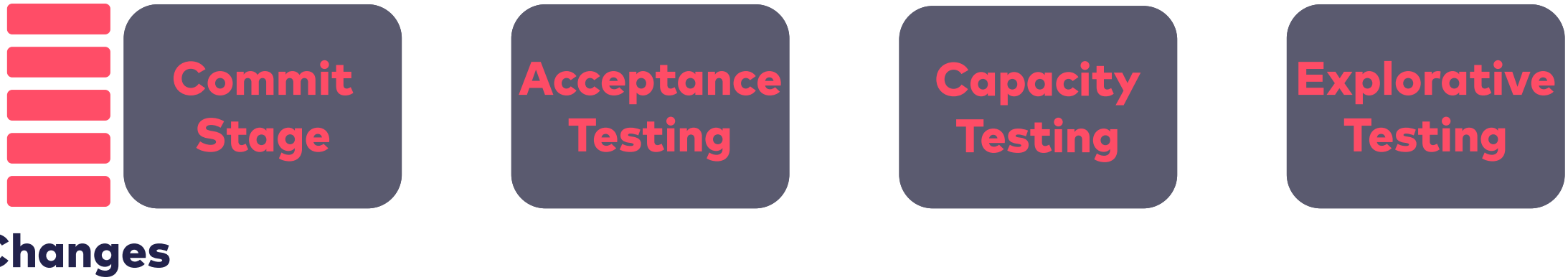
Feedback & Continuous Delivery

- Continuous Delivery:
  - Automate pipeline
  - Make feedback cheaper
  - Execute pipeline more often
  - Provide feedback more frequently

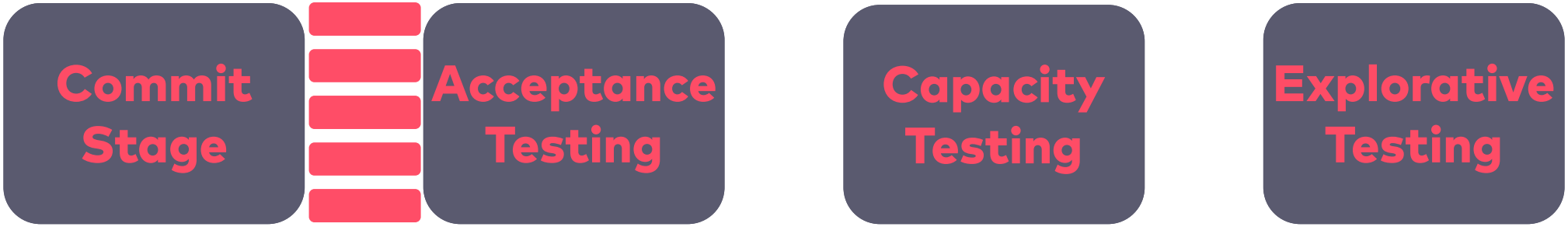
Lean

- Origin: Toyota production system
- Constant flow
- Identify bottlenecks

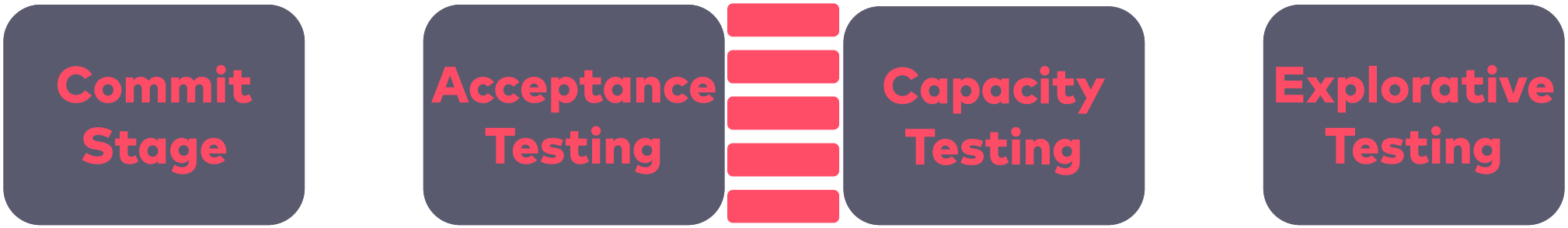
Without Continuous Delivery



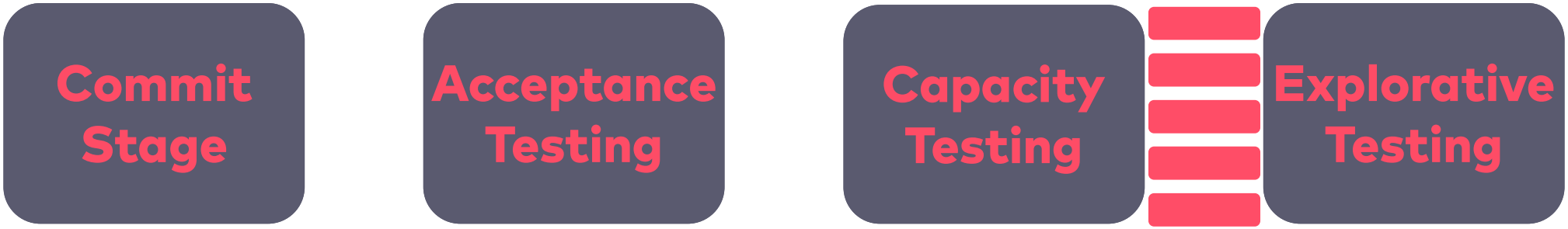
Without Continuous Delivery



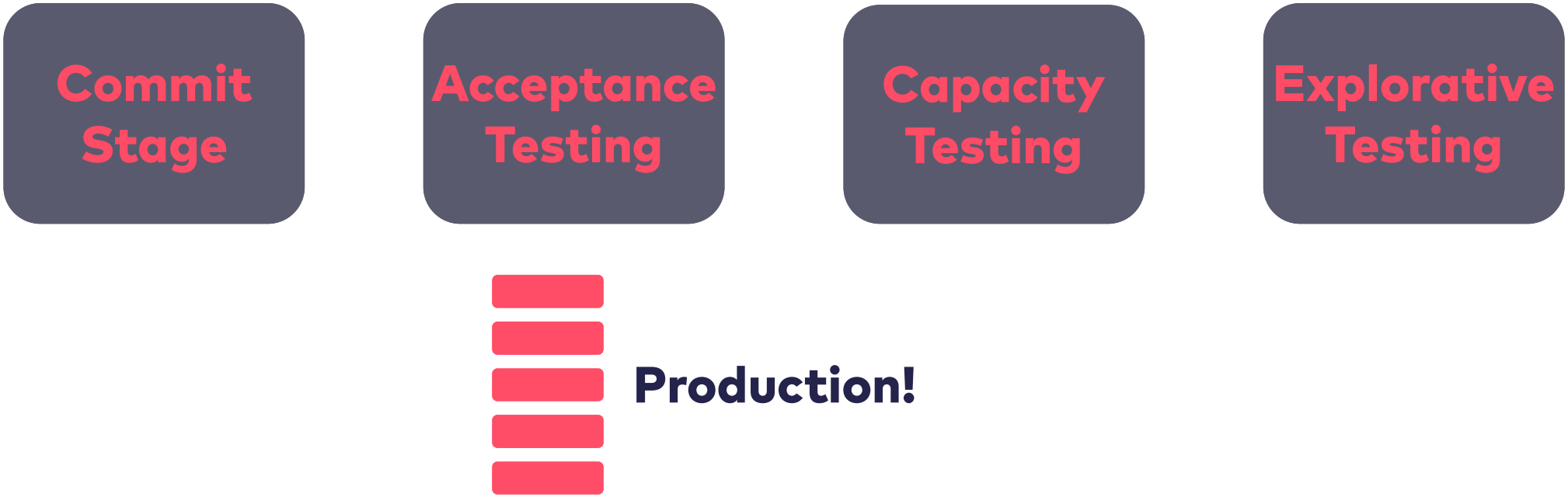
Without Continuous Delivery



Without Continuous Delivery



Without Continuous Delivery

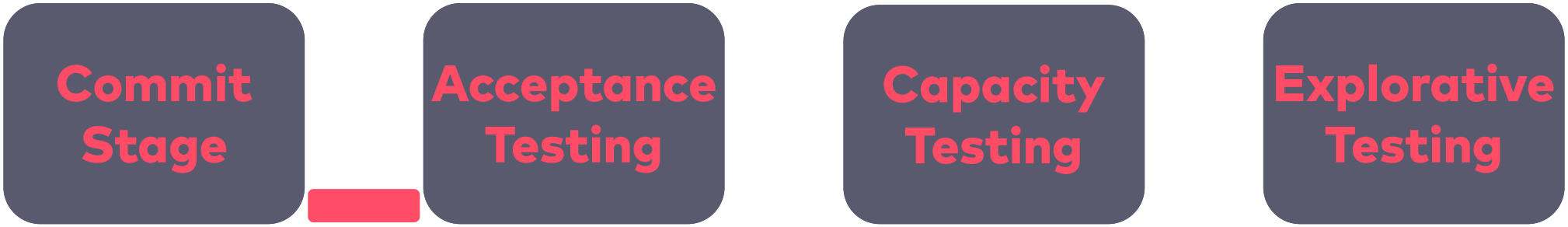


Continuous Delivery

- Increase speed
  - i.e. apply pressure to the pipeline
- Deploy changes independently and as fast as possible
- Aim for a constant flow

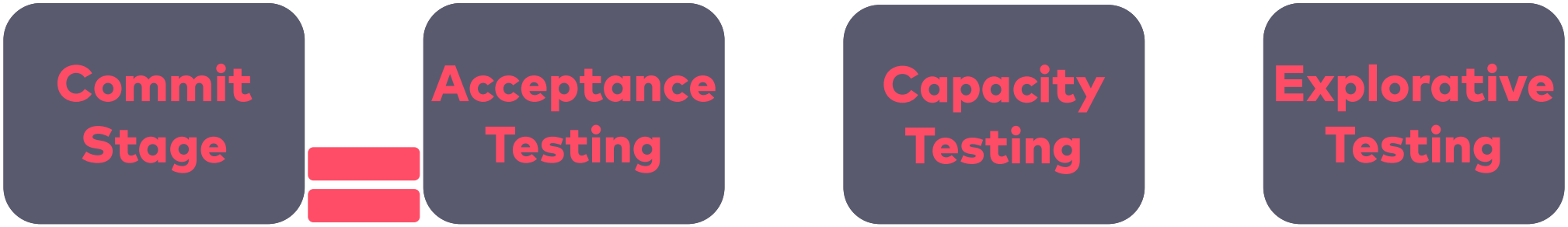
Continuous Delivery

Continuous Delivery



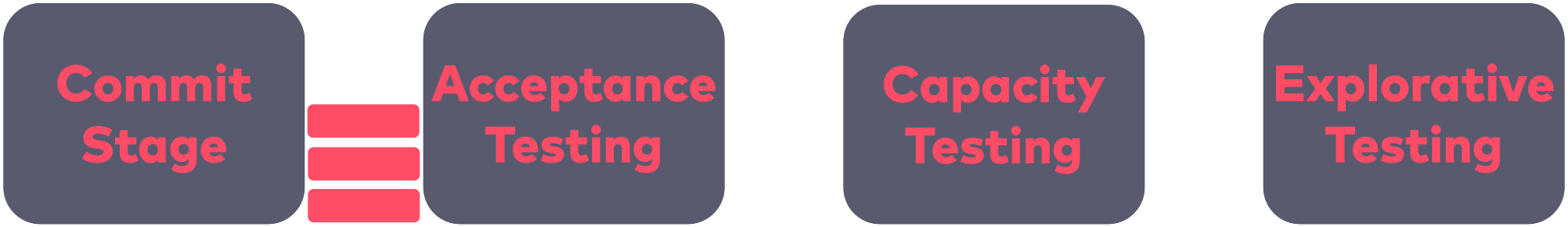
Continuous Delivery

Continuous Delivery



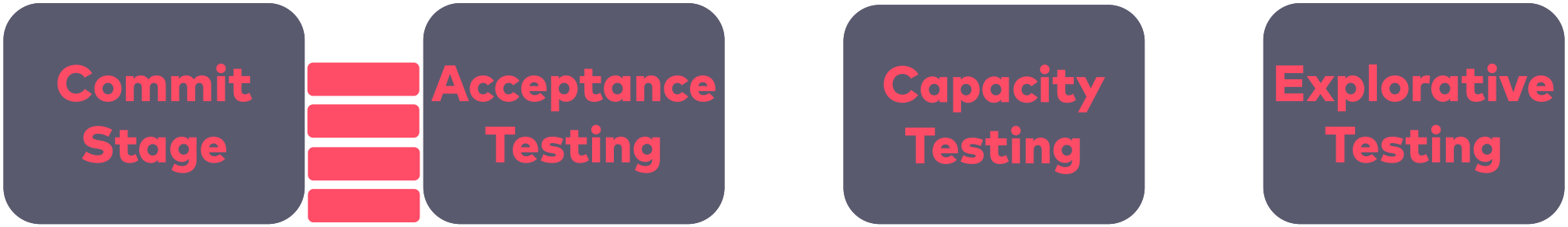
Continuous Delivery

Continuous Delivery

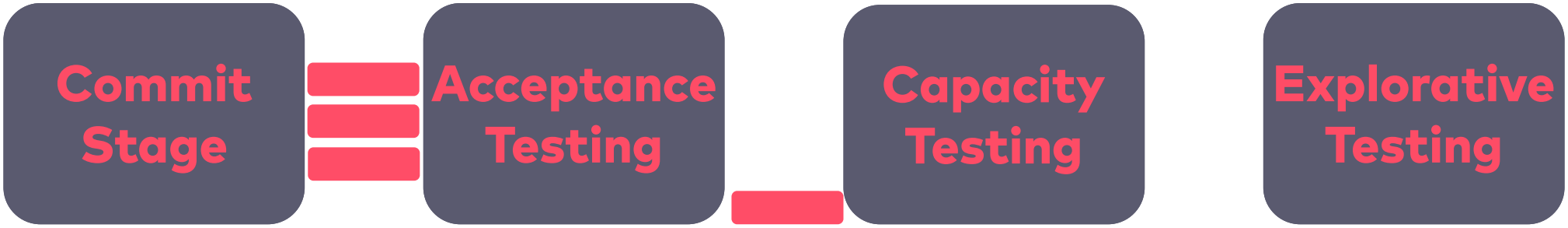


Continuous Delivery

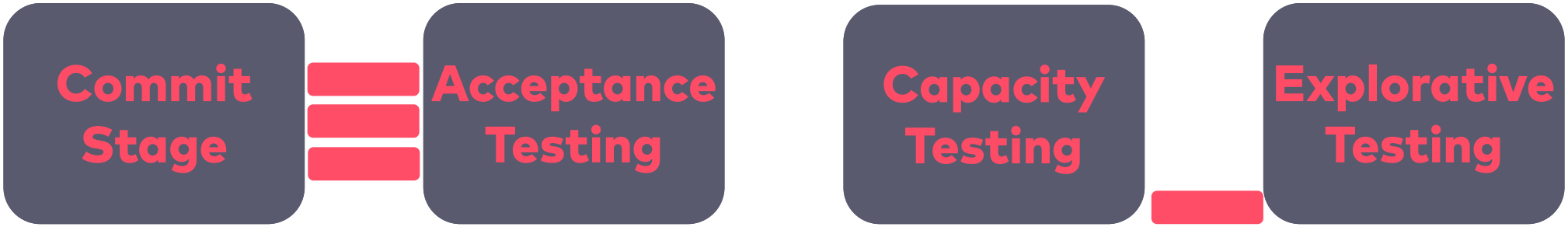
Continuous Delivery



Continuous Delivery

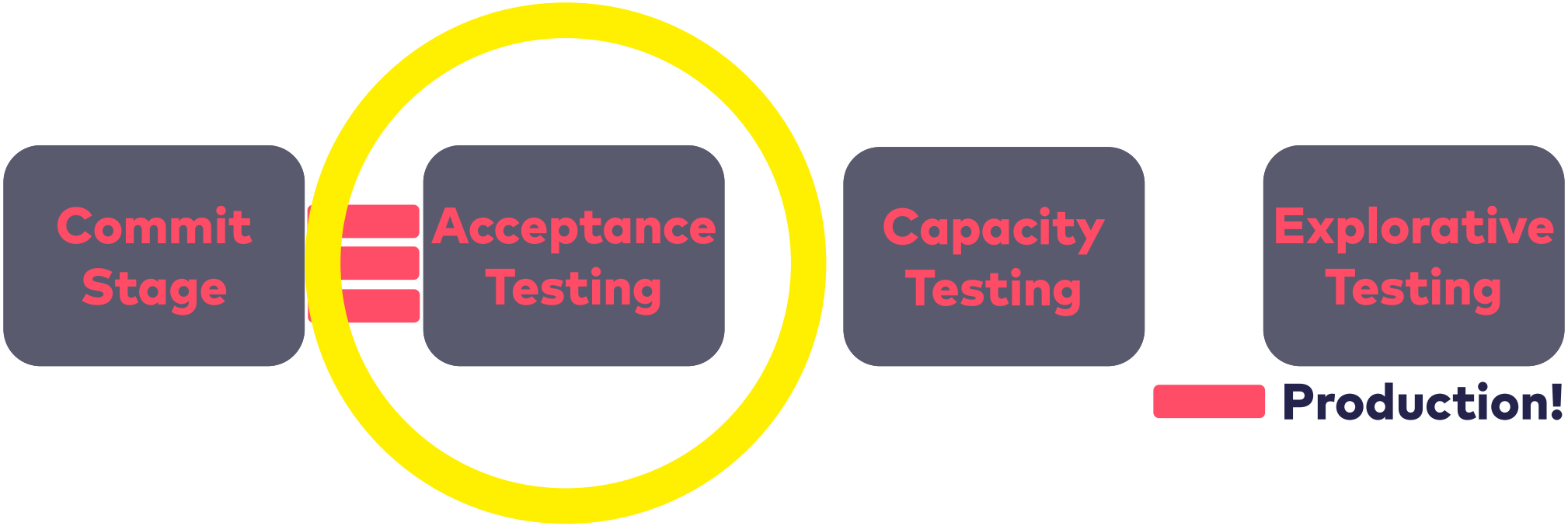


Continuous Delivery



Continuous Delivery

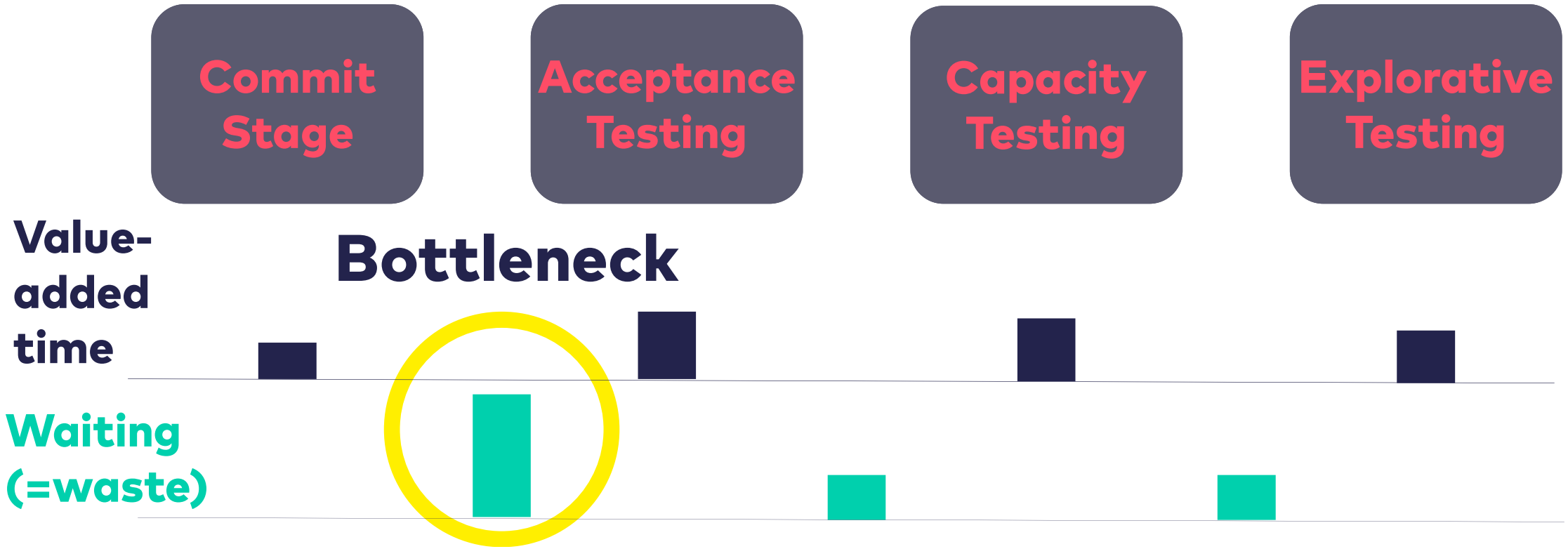
Continuous Delivery



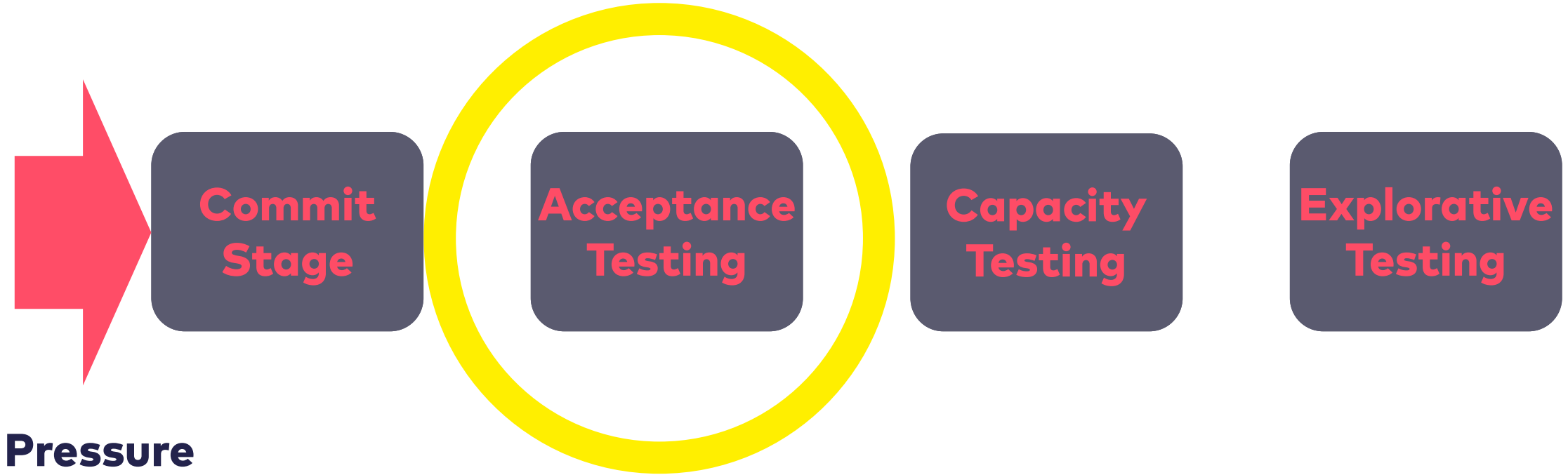
Bottleneck

Need to optimize acceptance test

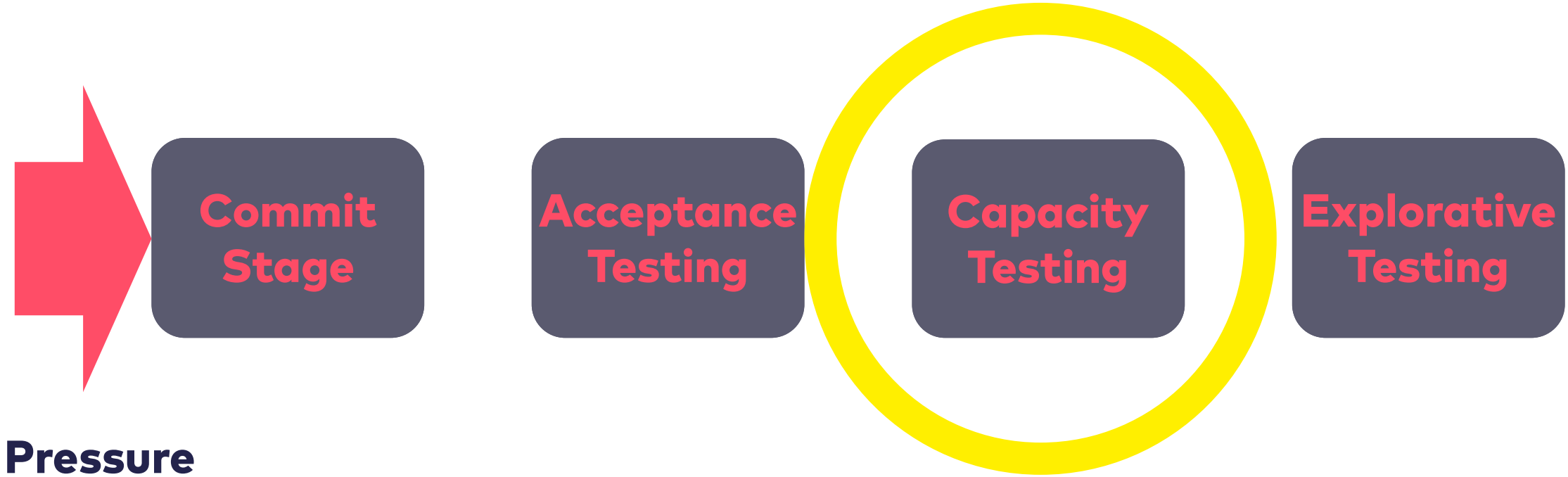
Value Stream Mapping



Identify and Eliminate Bottleneck



Identify and Eliminate Bottleneck



Continuous Delivery as Optimization

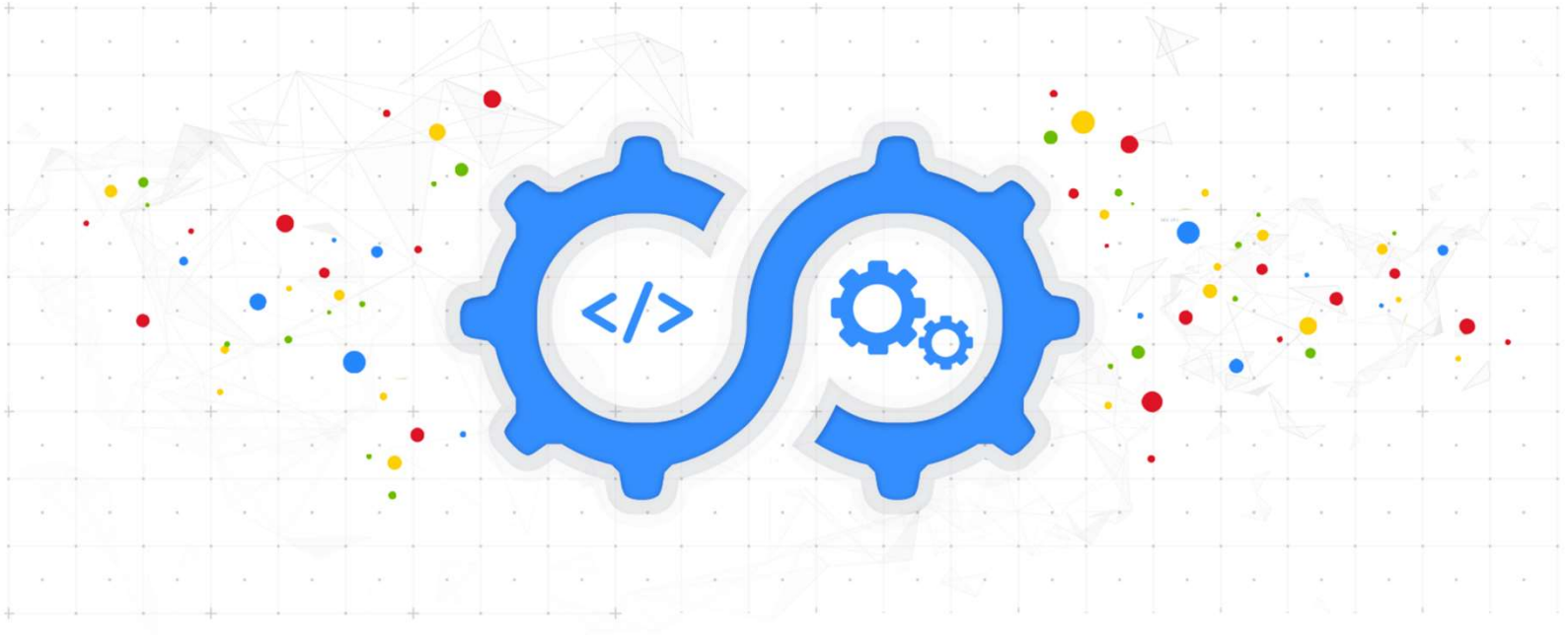
- Continuous delivery optimizes software development
- By increasing speed
- By focusing on software in production

Continuous Delivery as Optimization

- Makes bottlenecks obvious
- Helps to eliminate bottlenecks

- So there should be more advantages than time-to-market

The 2019 Accelerate State of DevOps: Elite performance, productivity, and scaling



https://cloud.google.com/devops/state-of-devops/

Deployment Frequency: Results

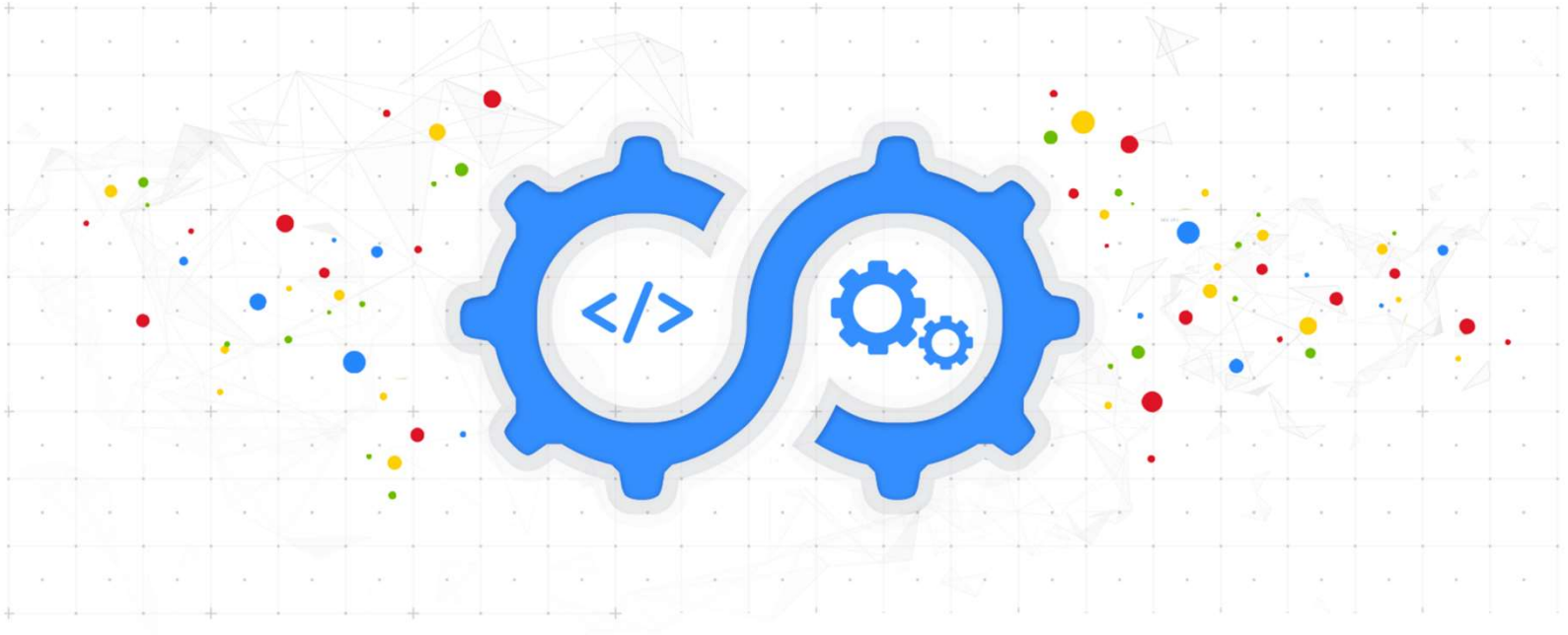
- Elite Performers vs. Low Performers
- Multiple times per day vs. once per month /6 months

- 106x better lead time for change
- 2.604x better time to restore service
- 7x better change failure rate
- 50% vs 30% time spent on new work (2018 report)
- Less work on security issues, bugs, end user support (2018 report)

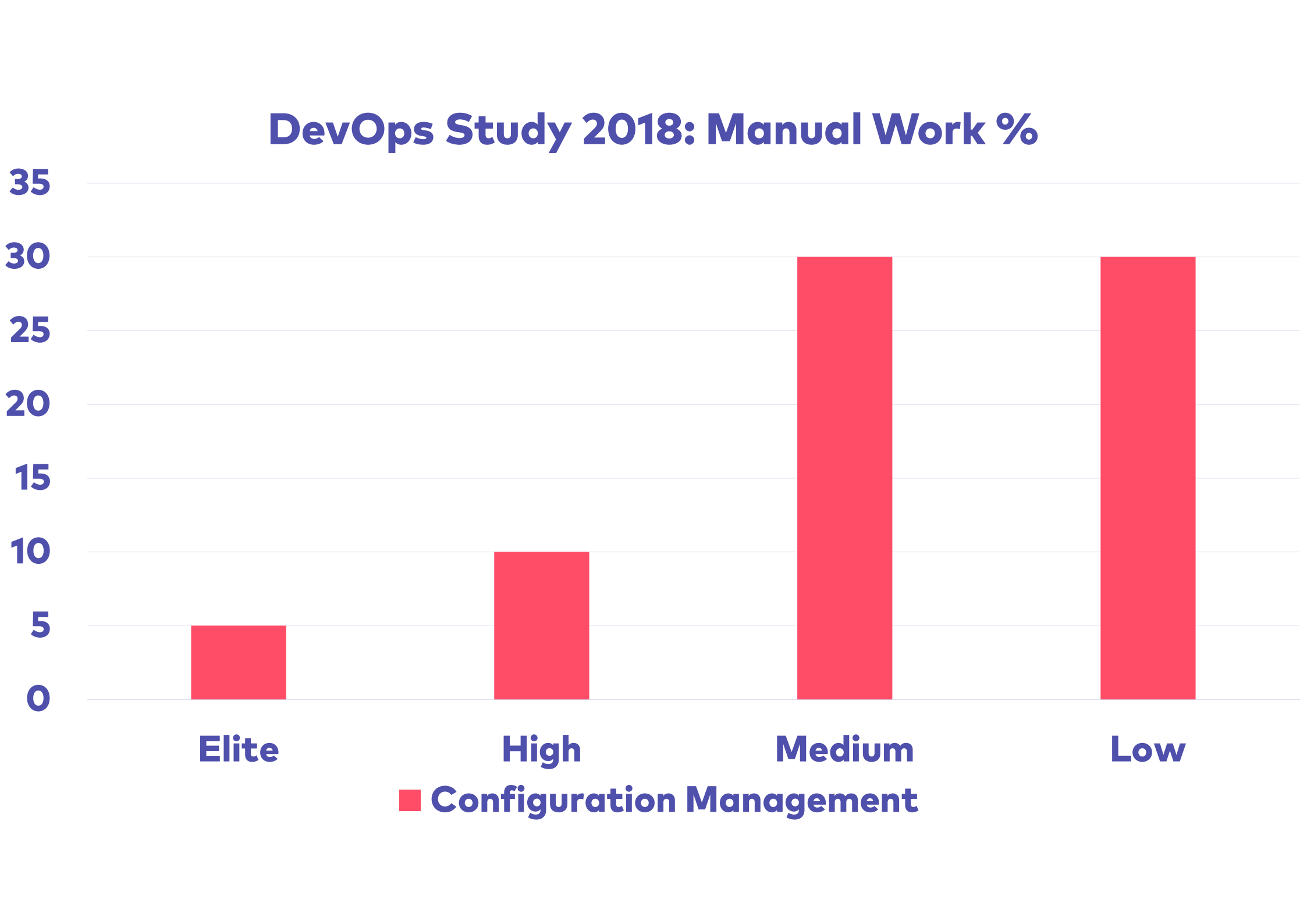
Continuous Delivery = Deployment Automation?

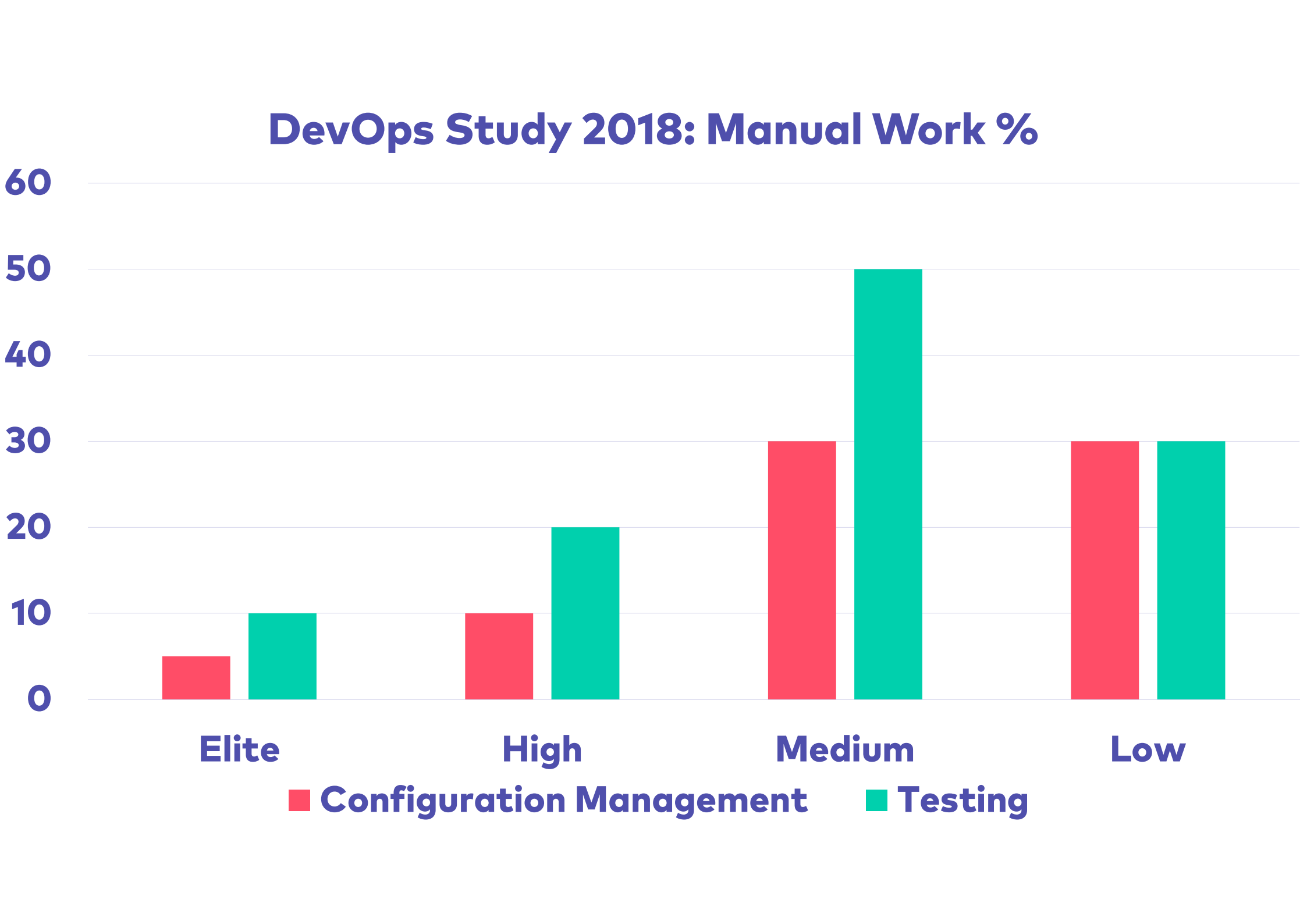
Where is the Deployment?

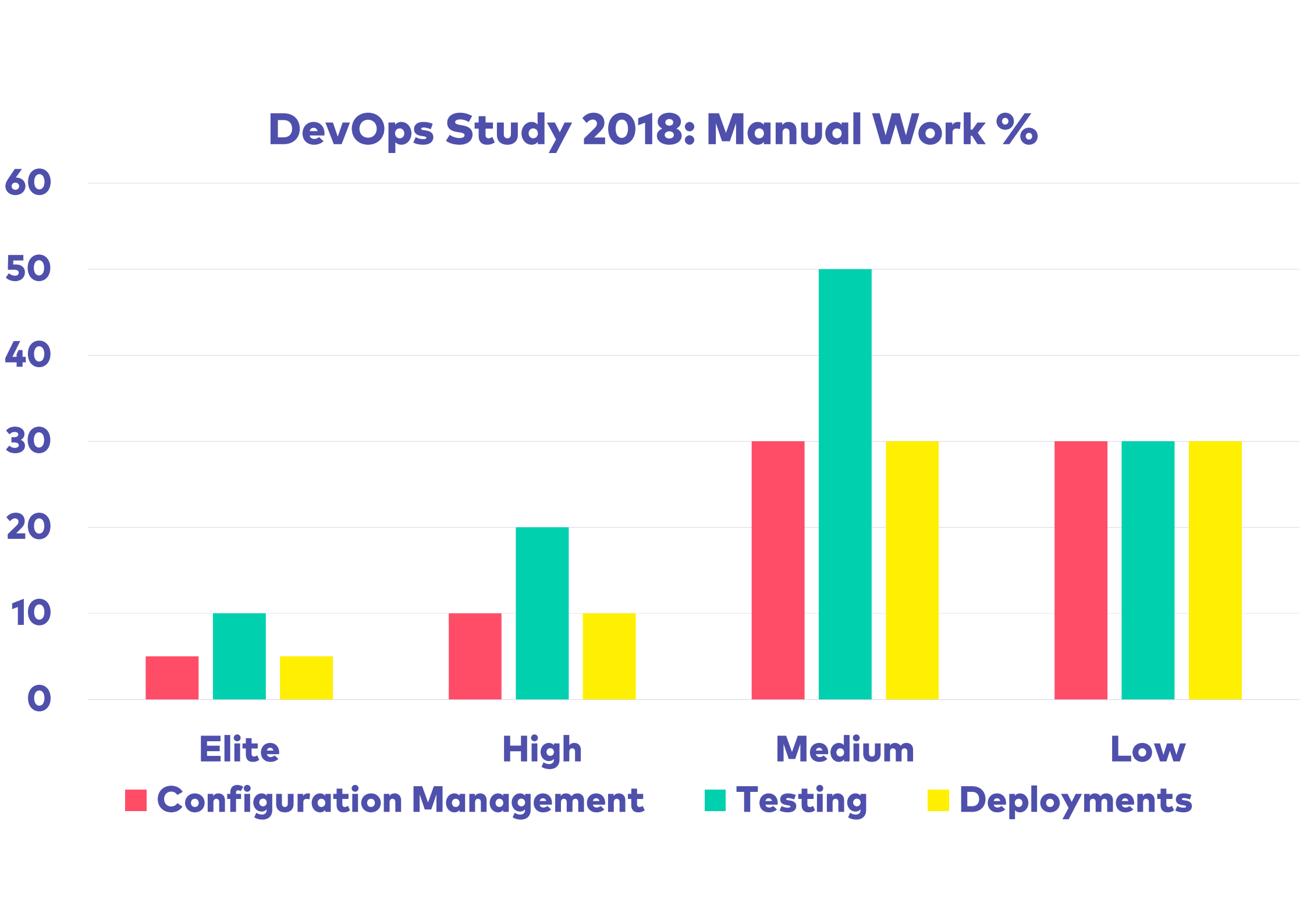
The 2019 Accelerate State of DevOps: Elite performance, productivity, and scaling



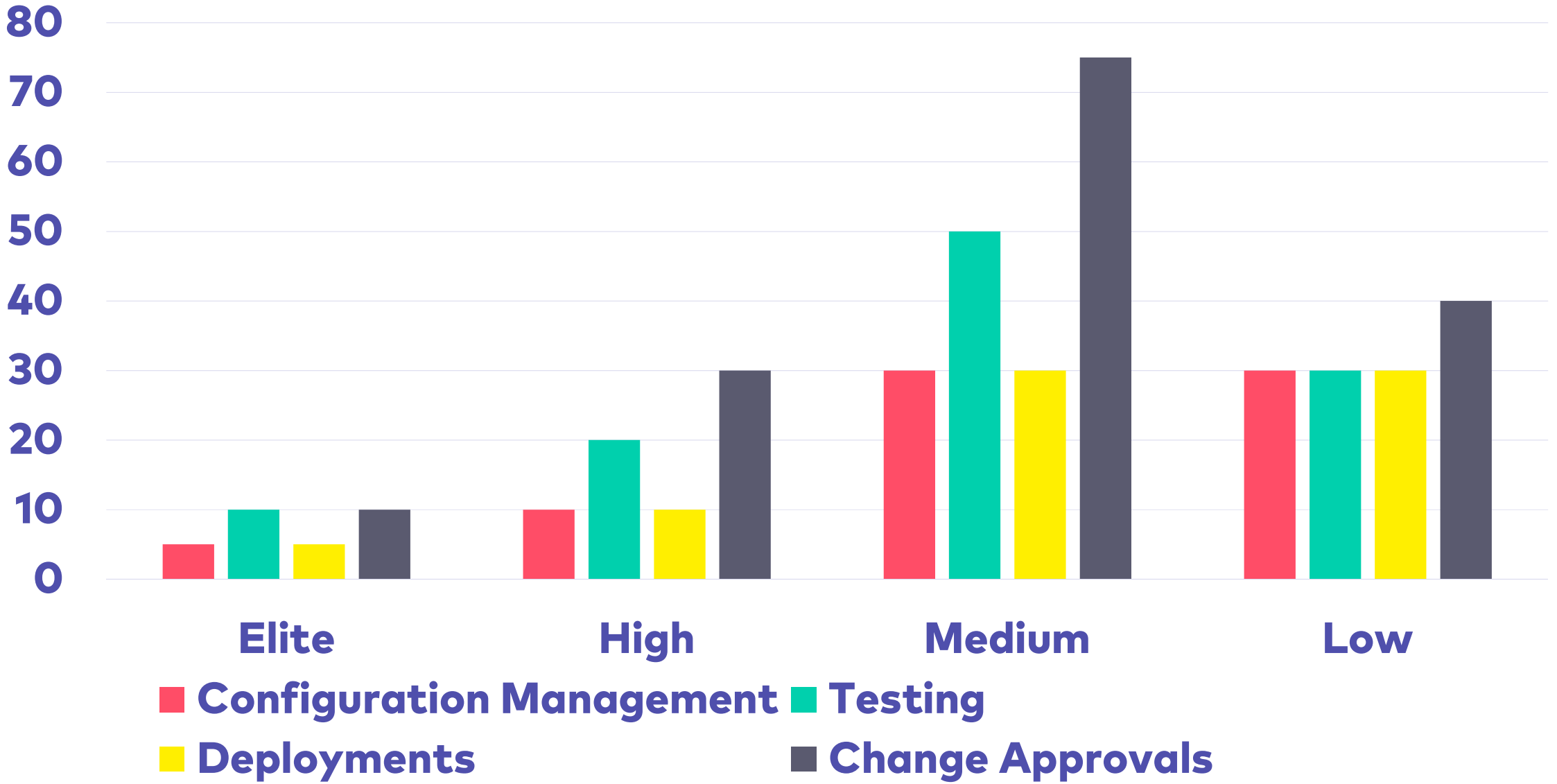
https://cloud.google.com/devops/state-of-devops/

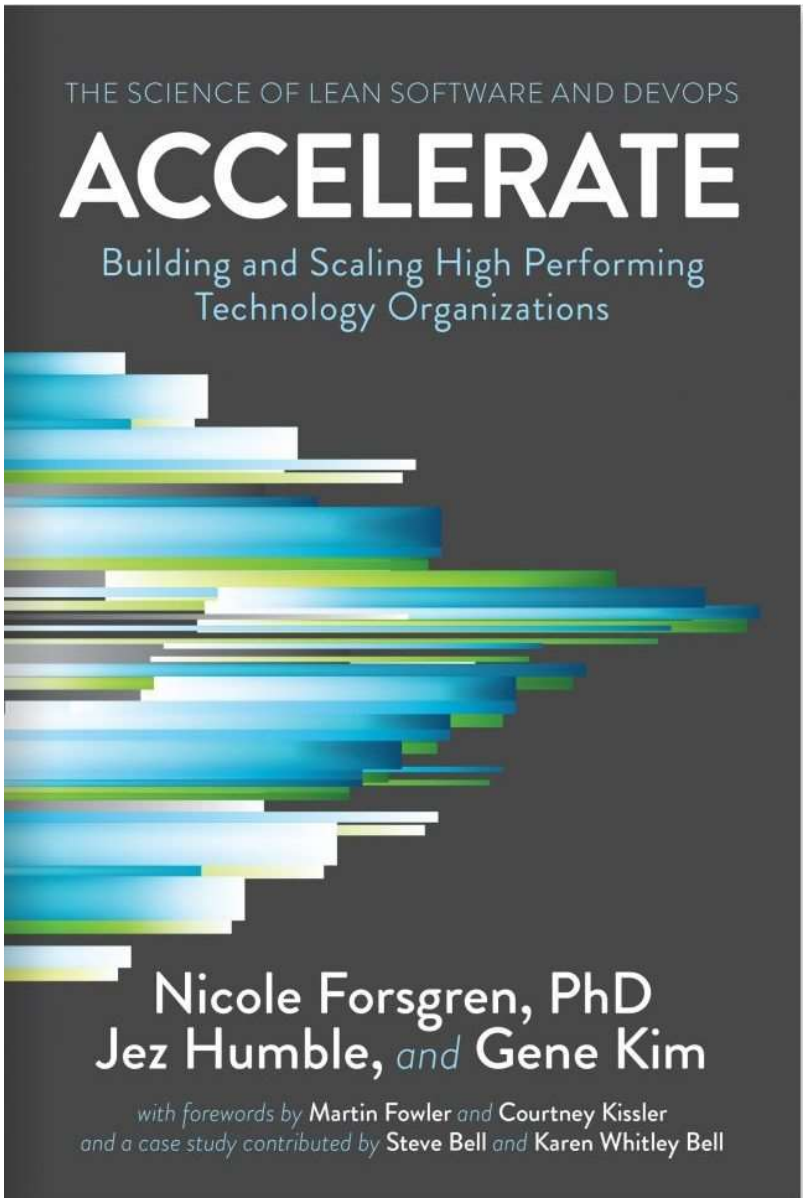

DevOps Study 2018: Manual Work %
35
30
25
20
15
10
5
0
Elite
High
Medium
Low
Configuration Management


DevOps Study 2018: Manual Work %
60
50
40
30
20
10
0
Elite
High
Medium
Low
Configuration Management
Testing


DevOps Study 2018: Manual Work %
60
50
40
30
20
10
0
Elite
High
Medium
Low
Configuration Management
Testing
Deployments

DevOps Study 2018: Manual Work %




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

Continuous Delivery as Optimization

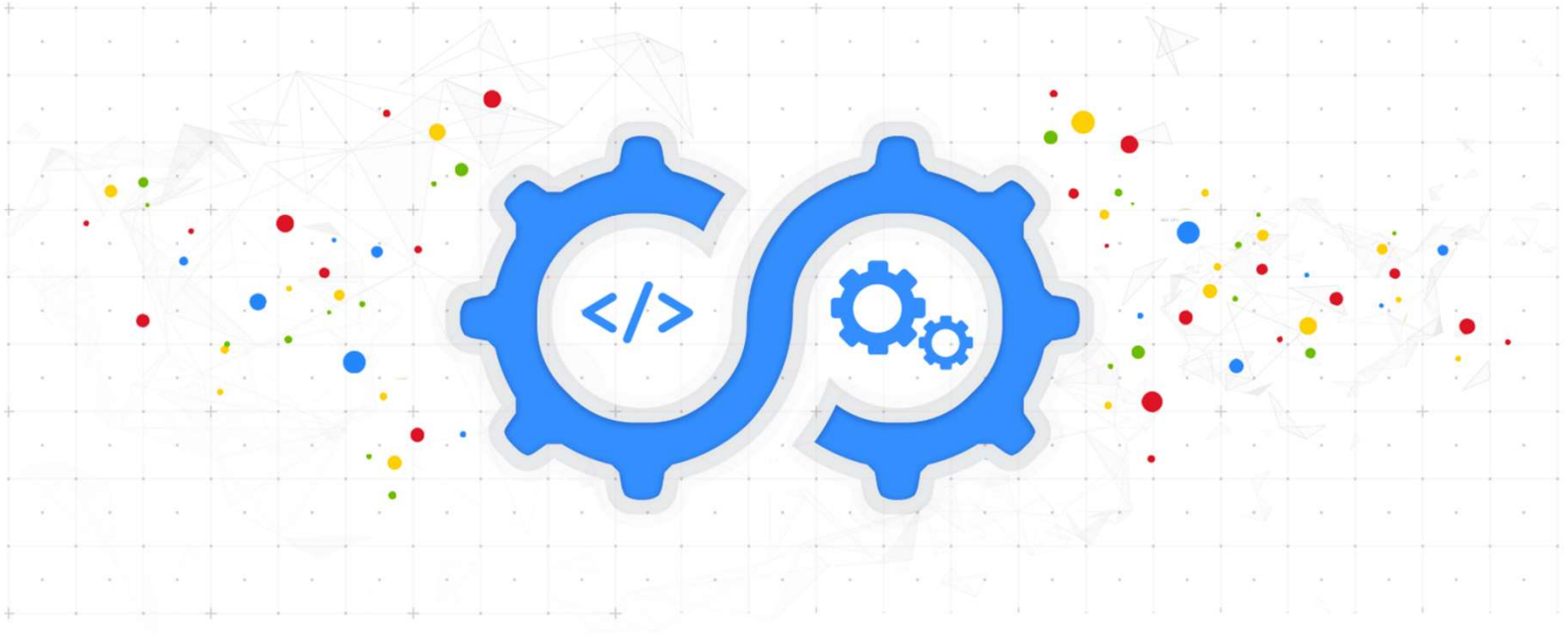
- Continuous Delivery causes a variety of optimizations
- Eliminate one bottleneck at a time!

Must deal with Continuous Delivery's feedback!

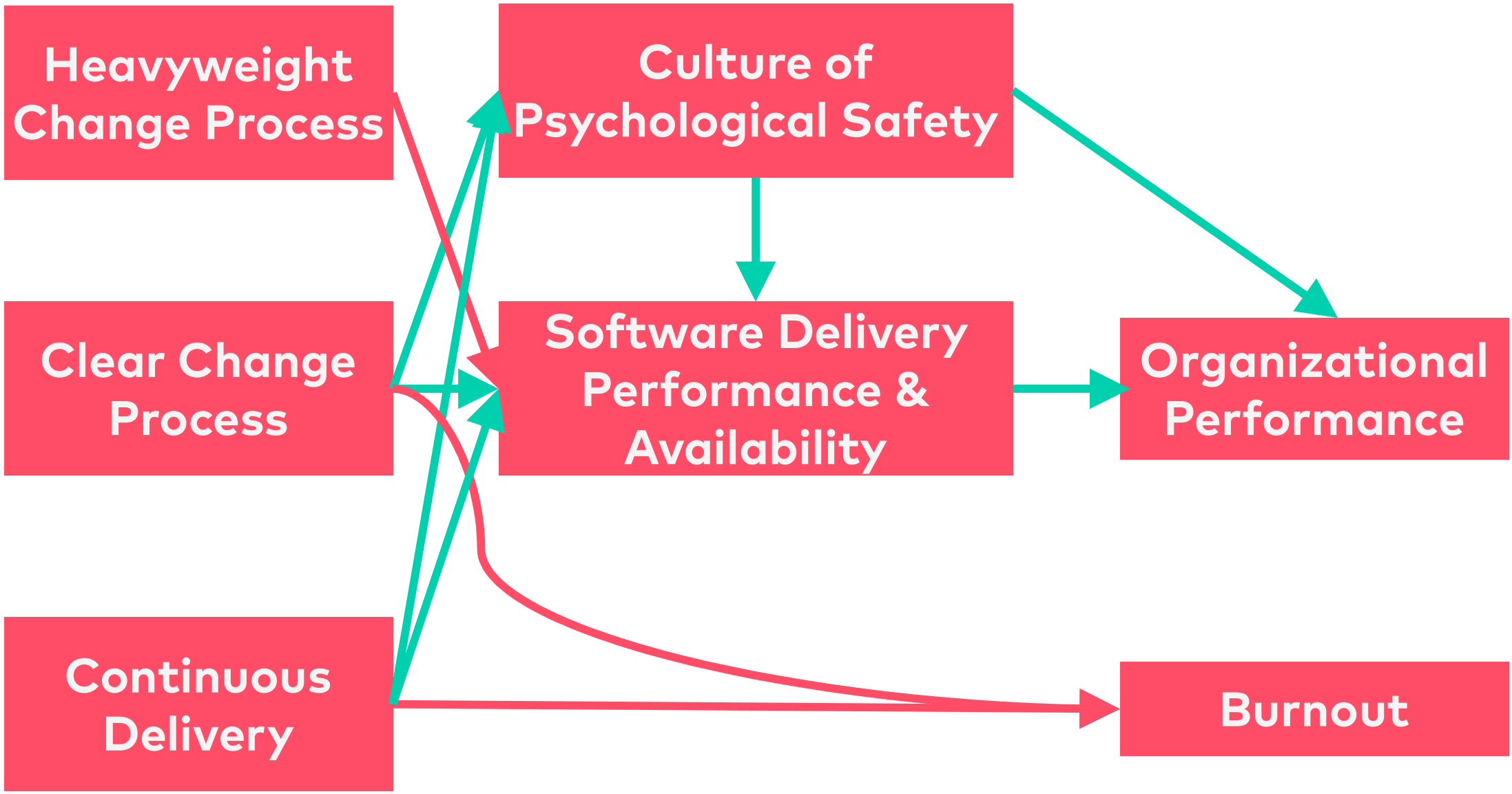
Must automate approval process!

More Results of Continuous Delivery

The 2019 Accelerate State of DevOps: Elite performance, productivity, and scaling



https://cloud.google.com/devops/state-of-devops/

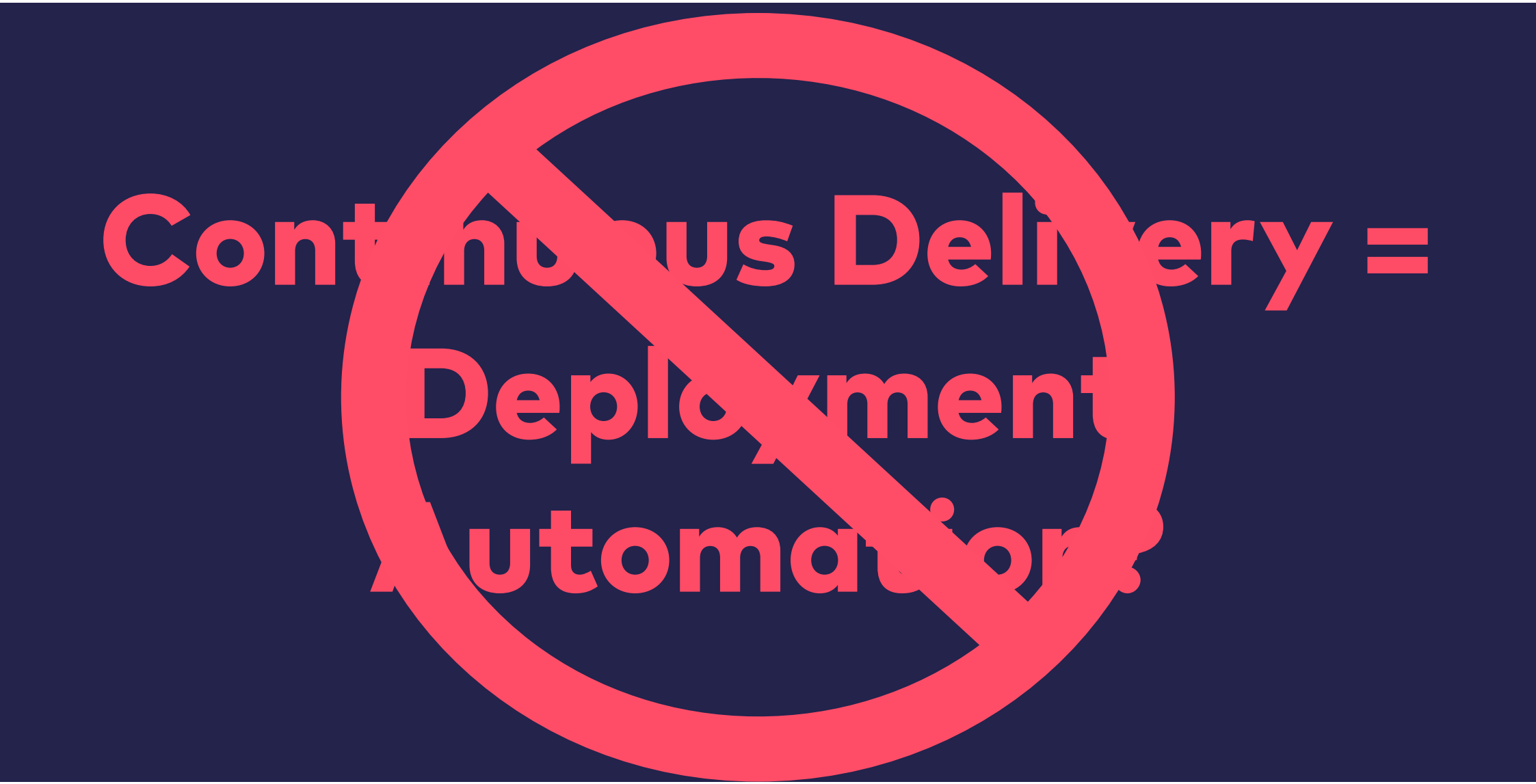

Heavyweight Change Process
Culture of Psychological Safety
Clear Change Process
Software Delivery Performance & Availability
Organizational Performance
Continuous Delivery
Burnout

More Results of Continuous Delivery

- Culture of Psychological Safety
- Less burnout
- Better organizational performance
  E.g. better market capitalization
- Employee net promoter score

- Results of empiric study

Continuous Delivery = Deployment Automation?


Continuous Delivery =
Deployment
Automation

Continuous Delivery = Time to Market


Continuous Delivery =
Time to Market

Continuous Delivery = Lean Optimization

Features = Release is a Risk

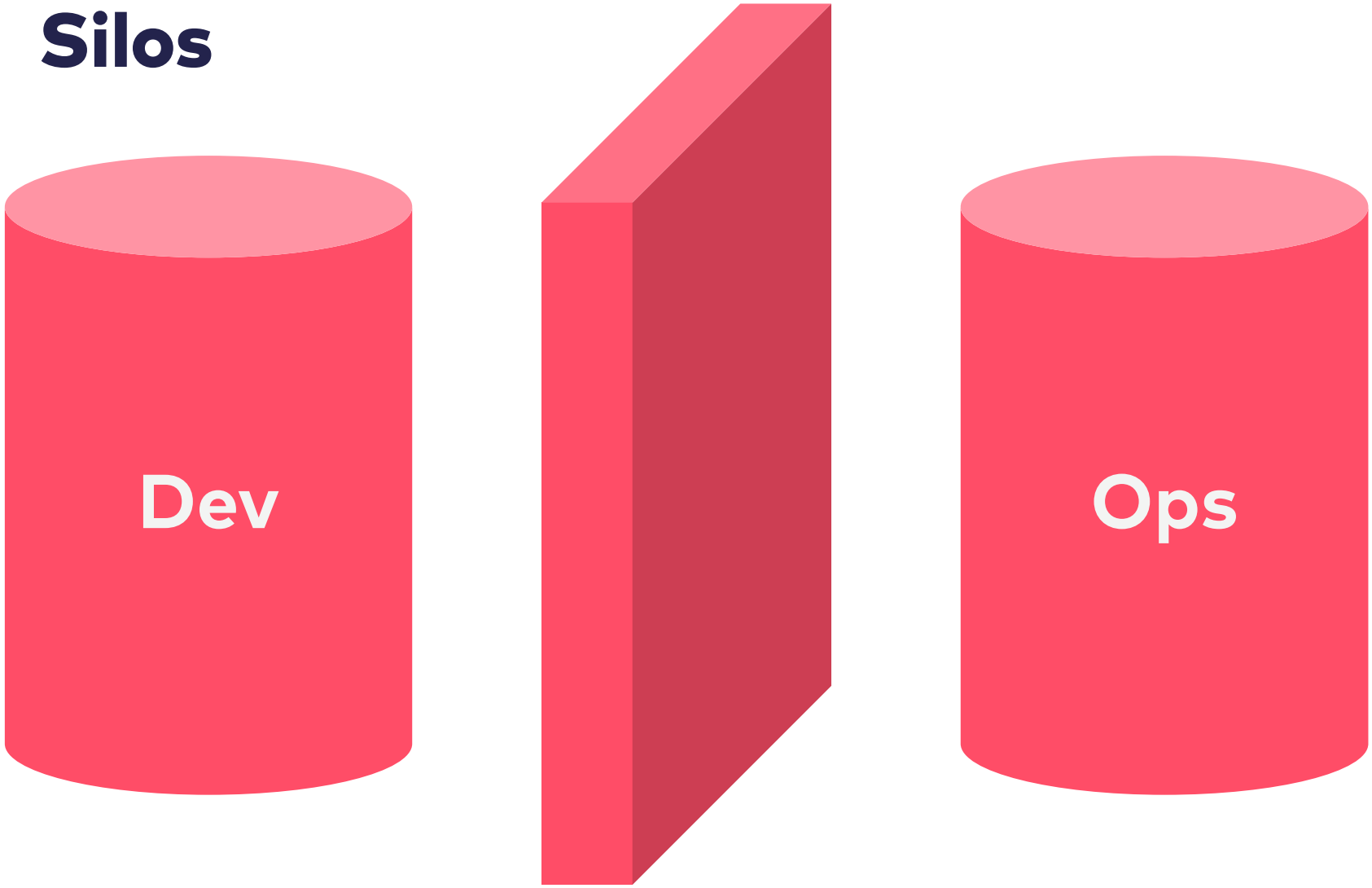
- Features have a deadline
- Sometimes deadlines are hard
- E.g. regulations


Release is a Risk
deadline
nes are hard
.g. regulations

Frequent Deployments Decrease Risk

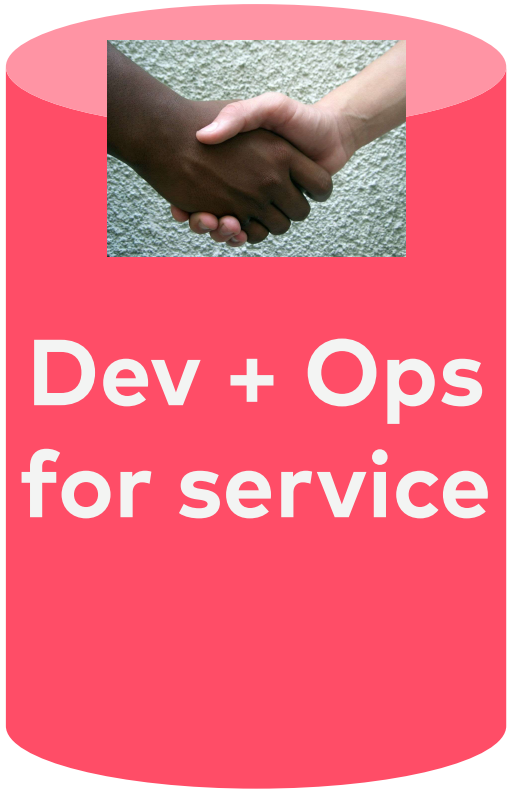
DevOps =
DevOps Engineer

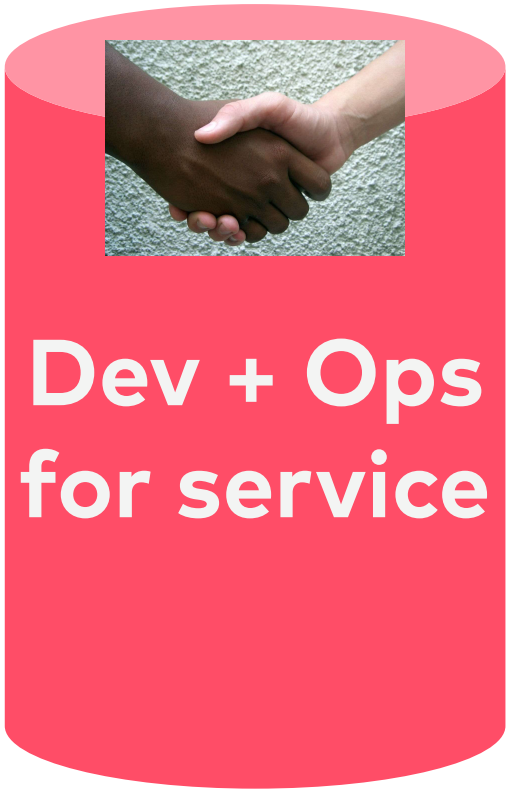
Silos



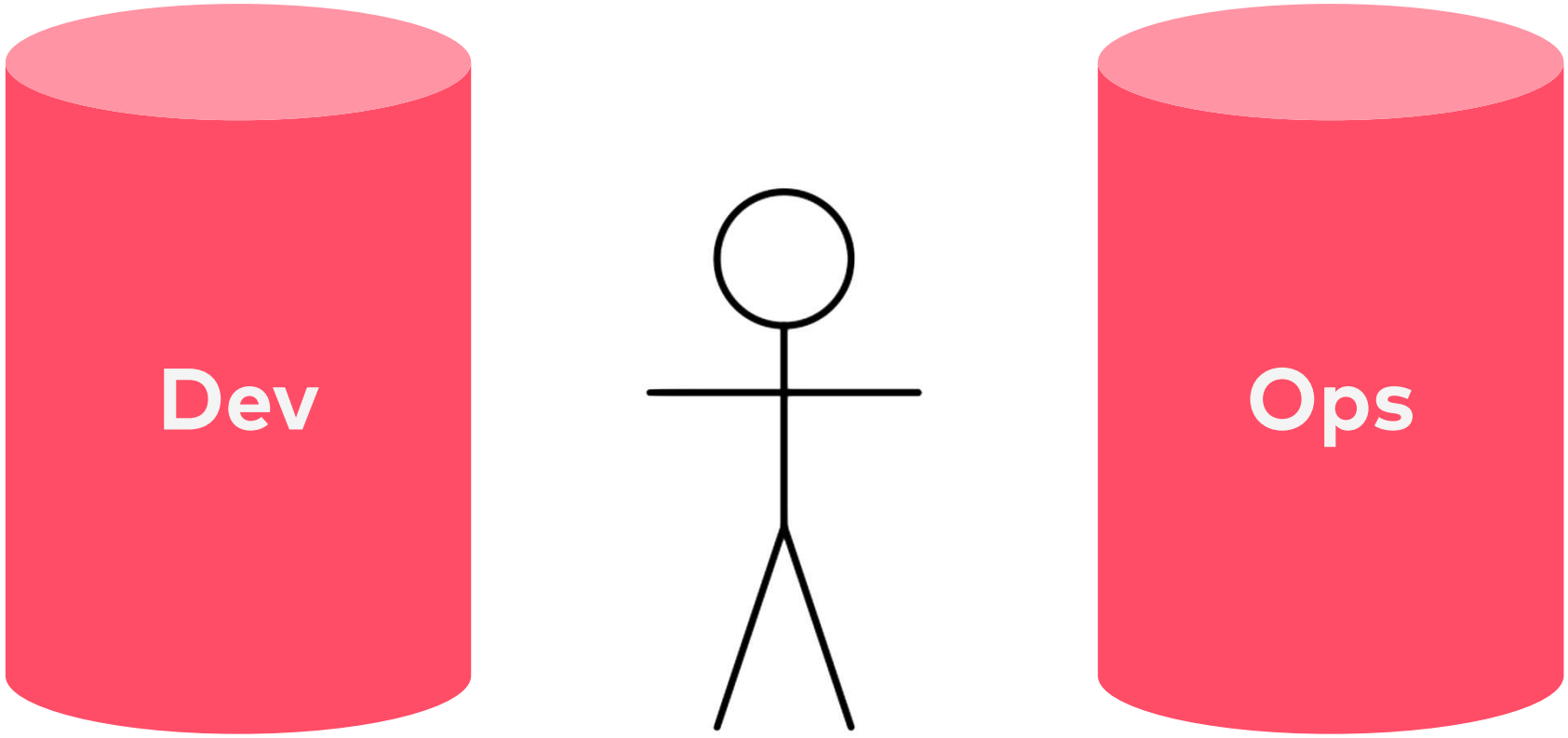
DevOps: Collaboration

DevOps: You Build It – You Run It



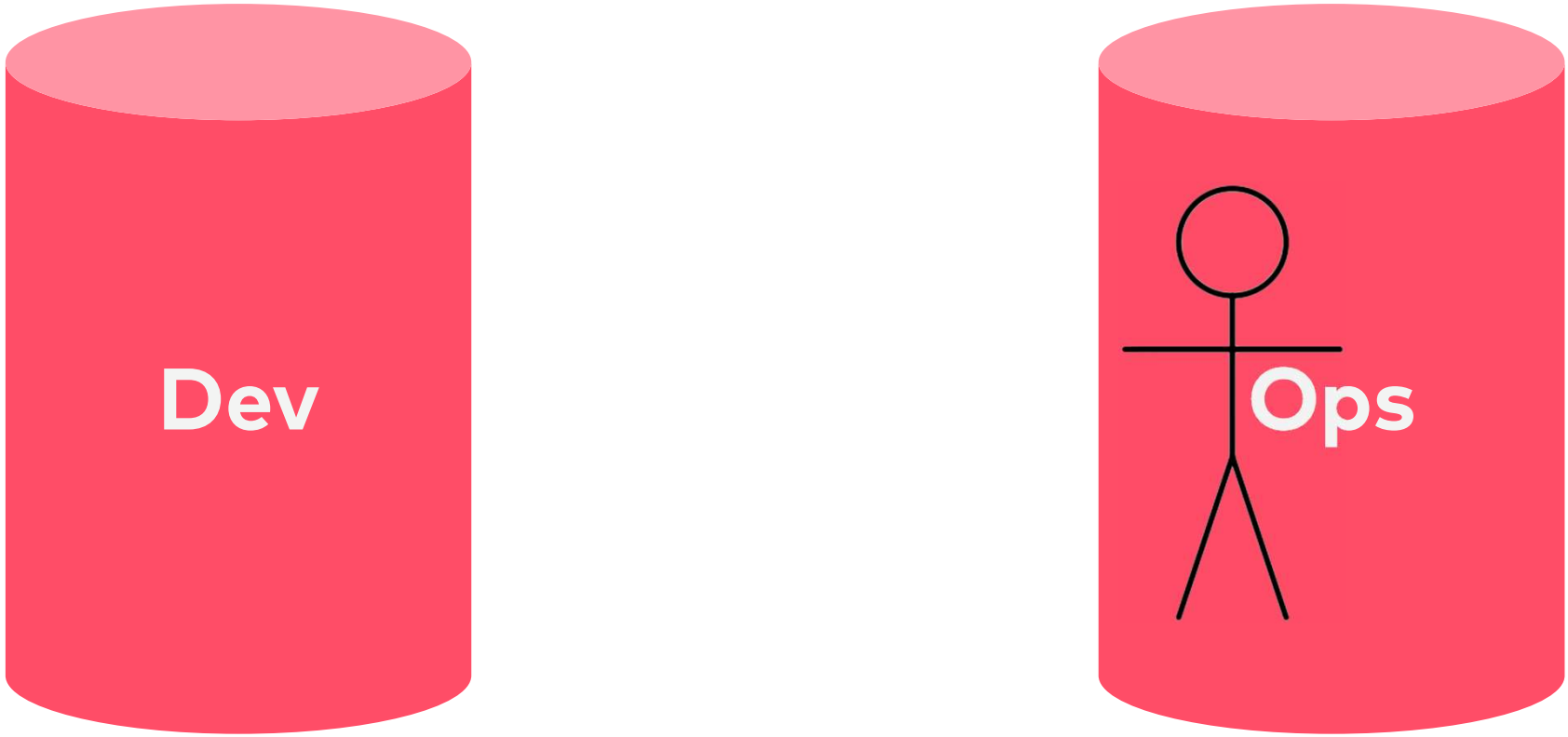


DevOps Engineer

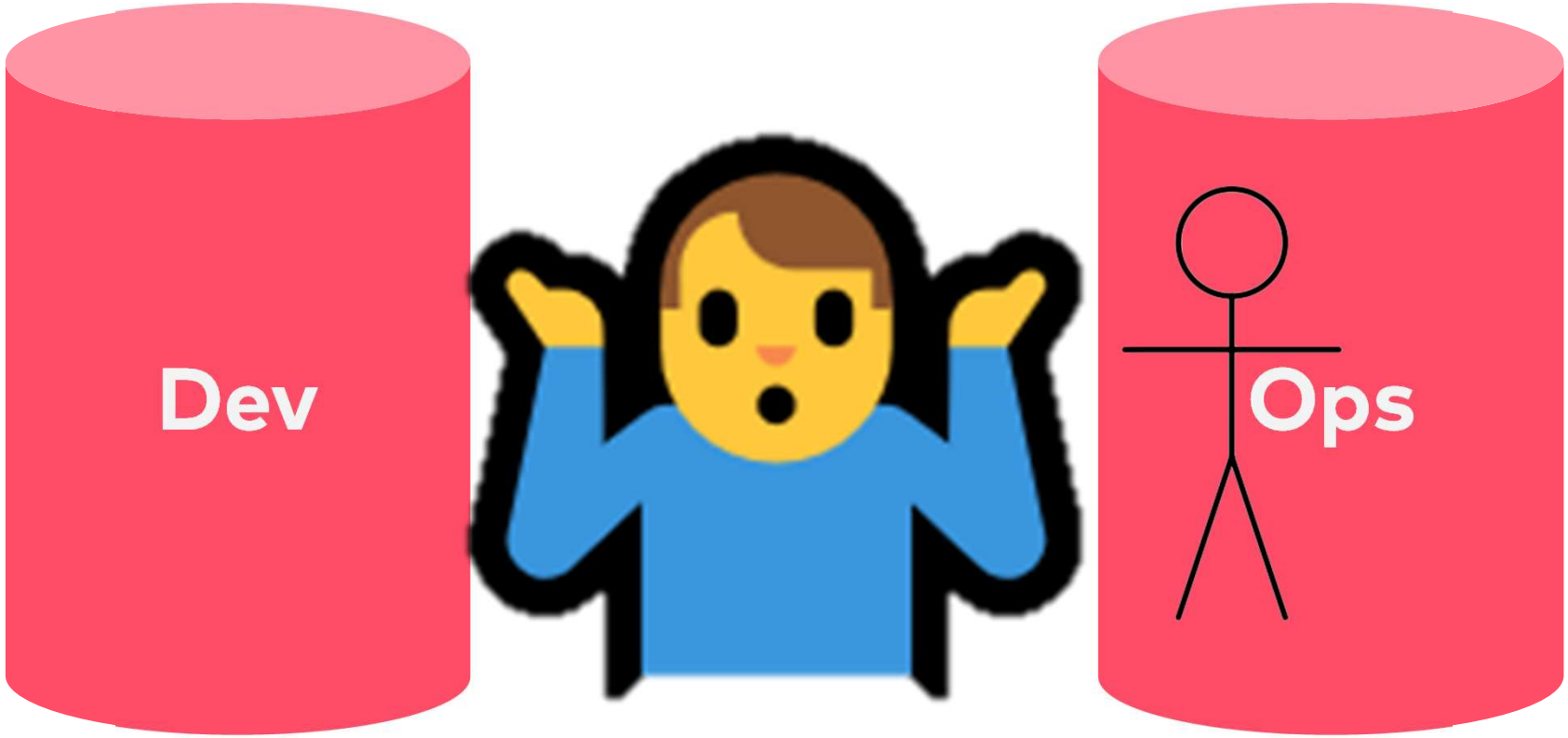


DevOps Engineer

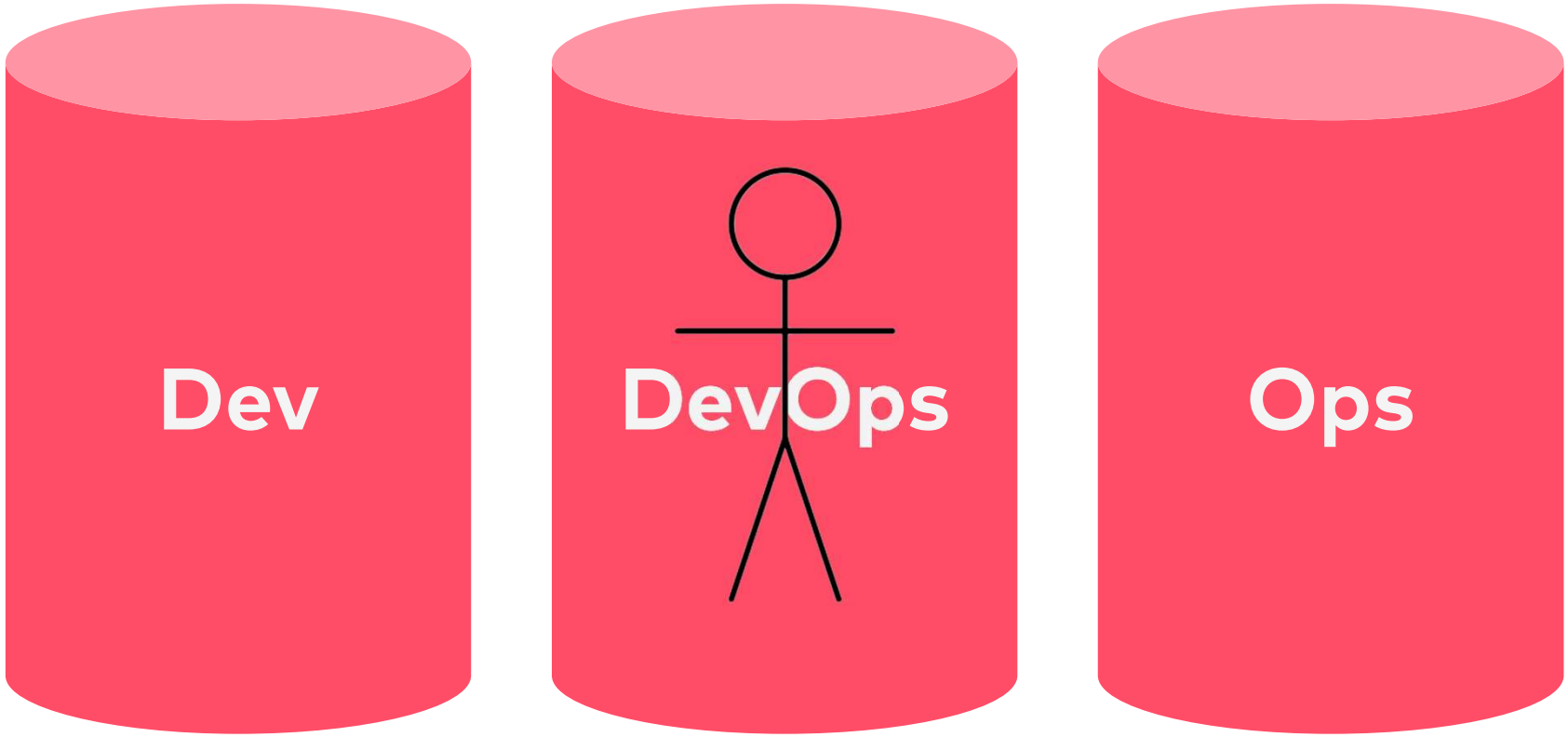
DevOps Engineer



DevOps Engineer

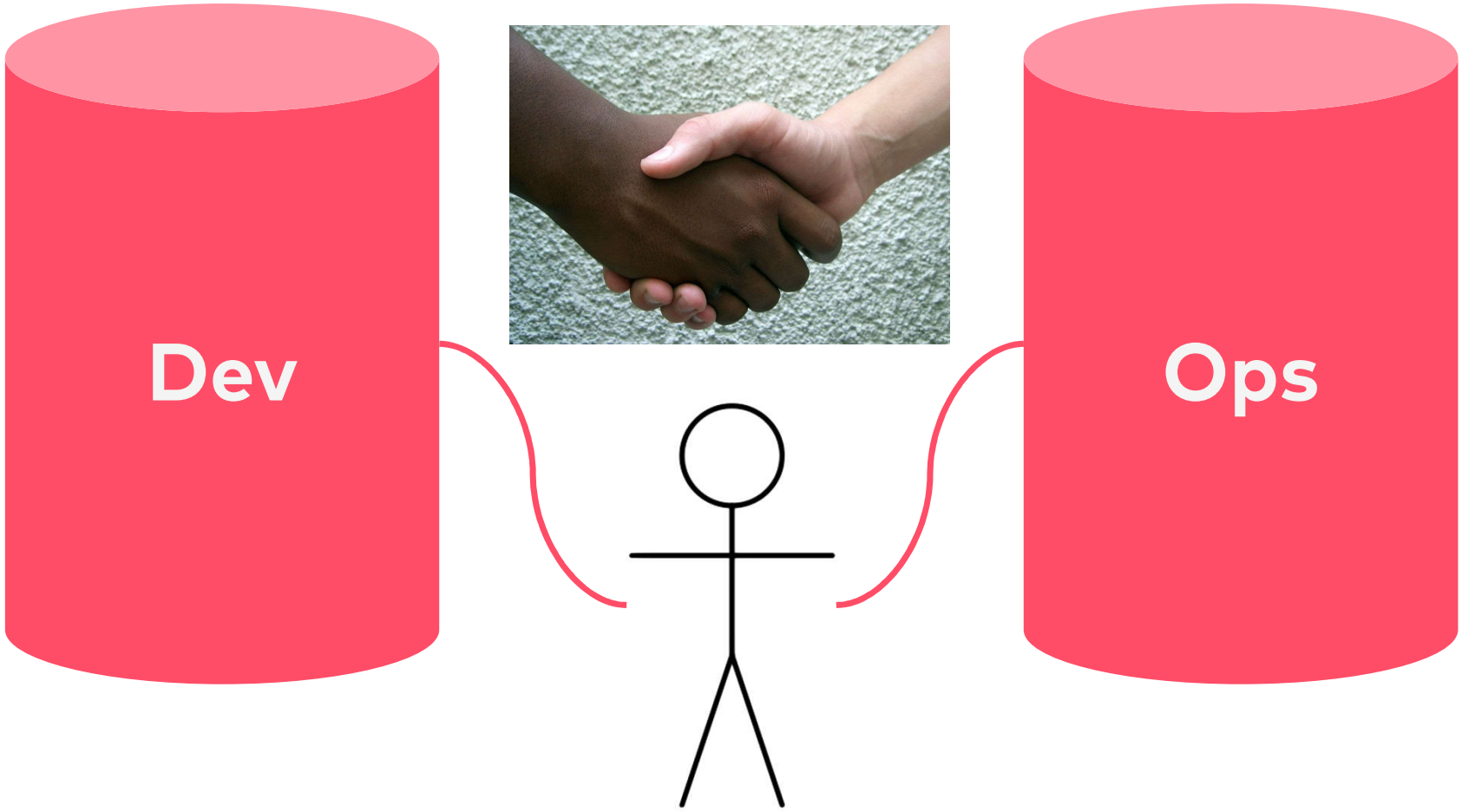


DevOps Engineer



DevOps Engineer

DevOps: Collaboration



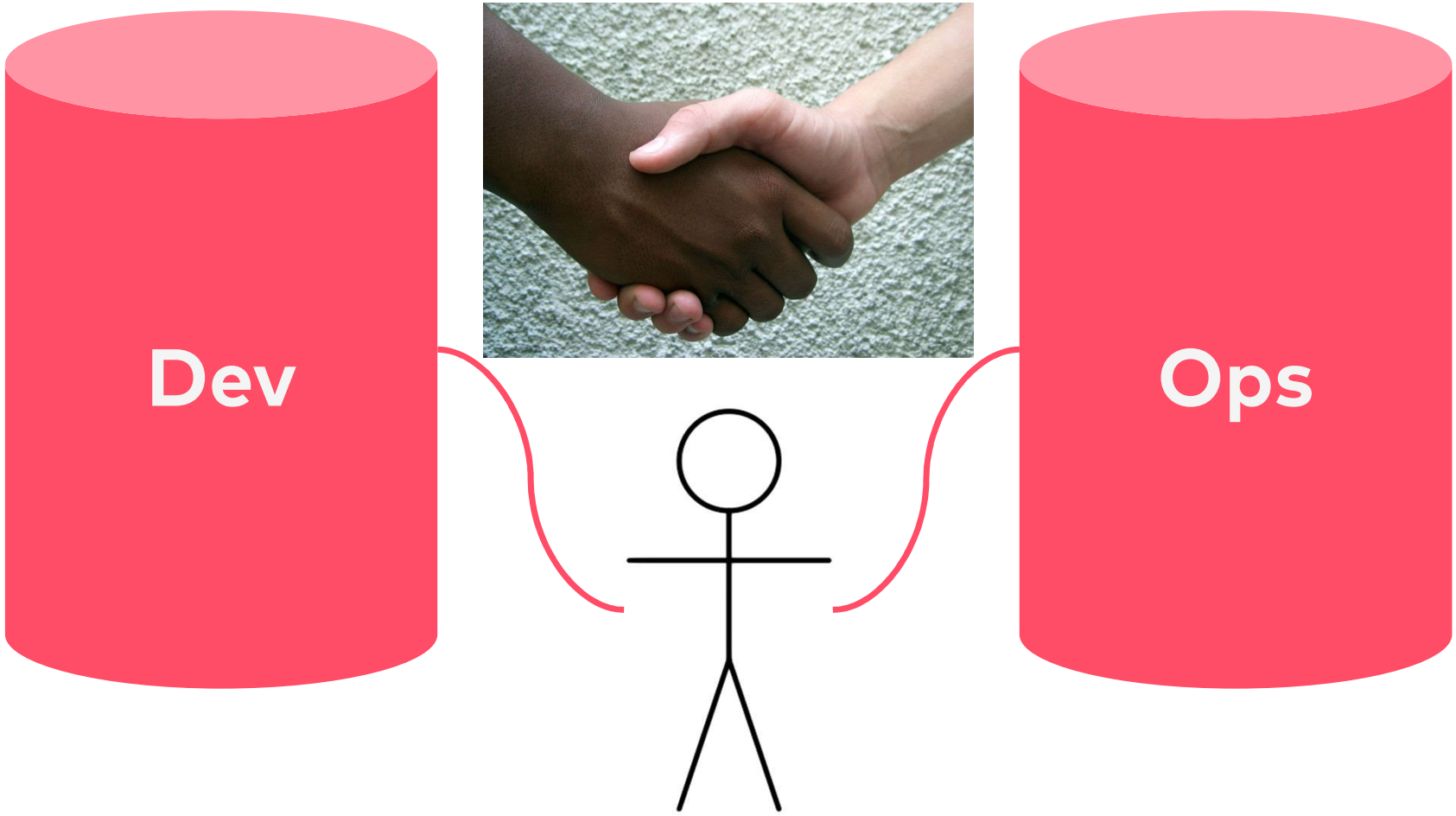
DevOps Engineer

- DevOps = collaboration
- Continuous delivery pipeline = shared artifact
- Continuous delivery needs at least some collaboration

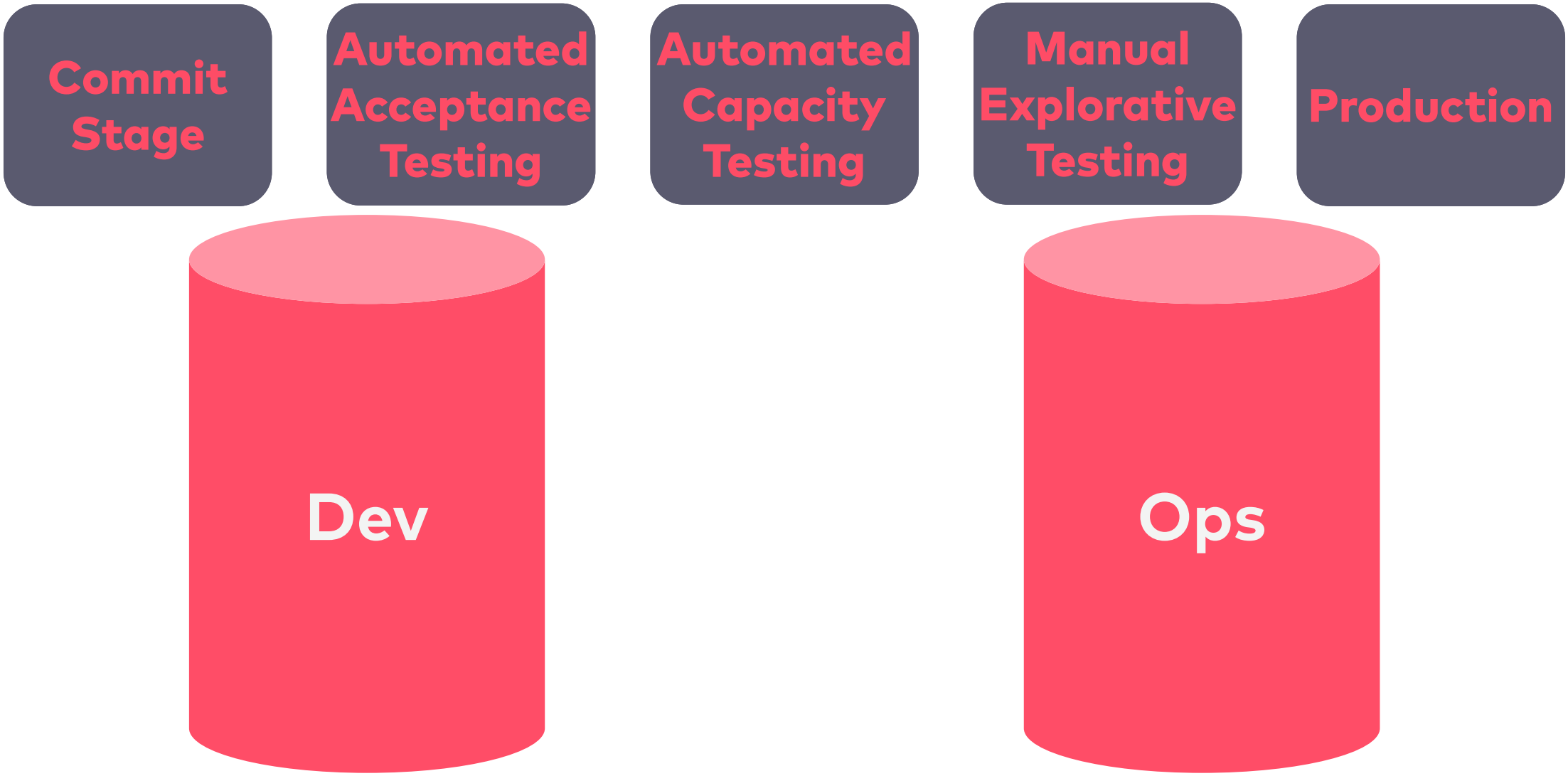
DevOps Engineer

- DevOps Engineer might make situation worse
- You need to transform the silos
- You need to ensure collaboration
- Mutual respect
- Incentive collaboration
- Cultural shift (sorry)

DevOps: Collaboration

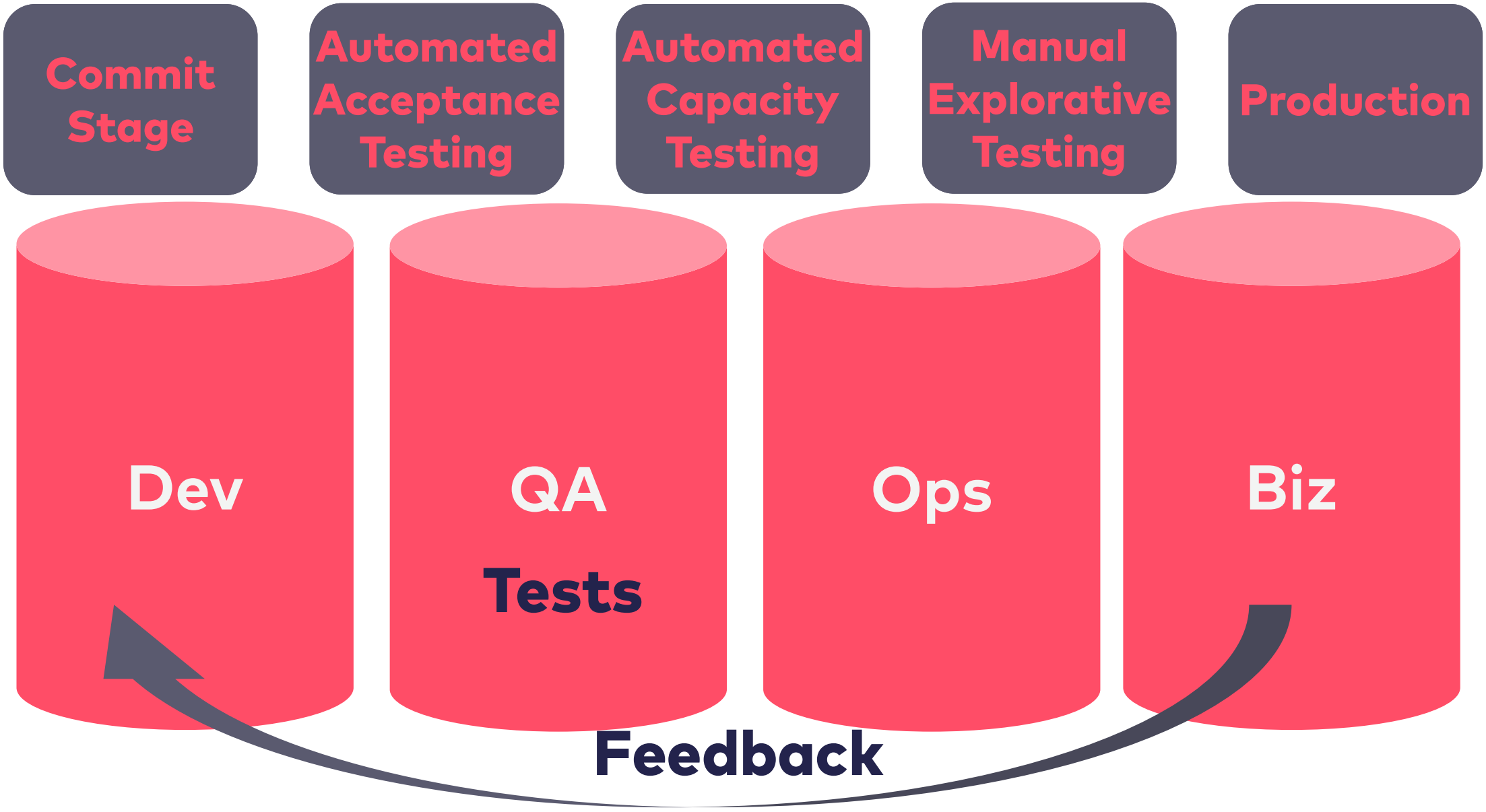


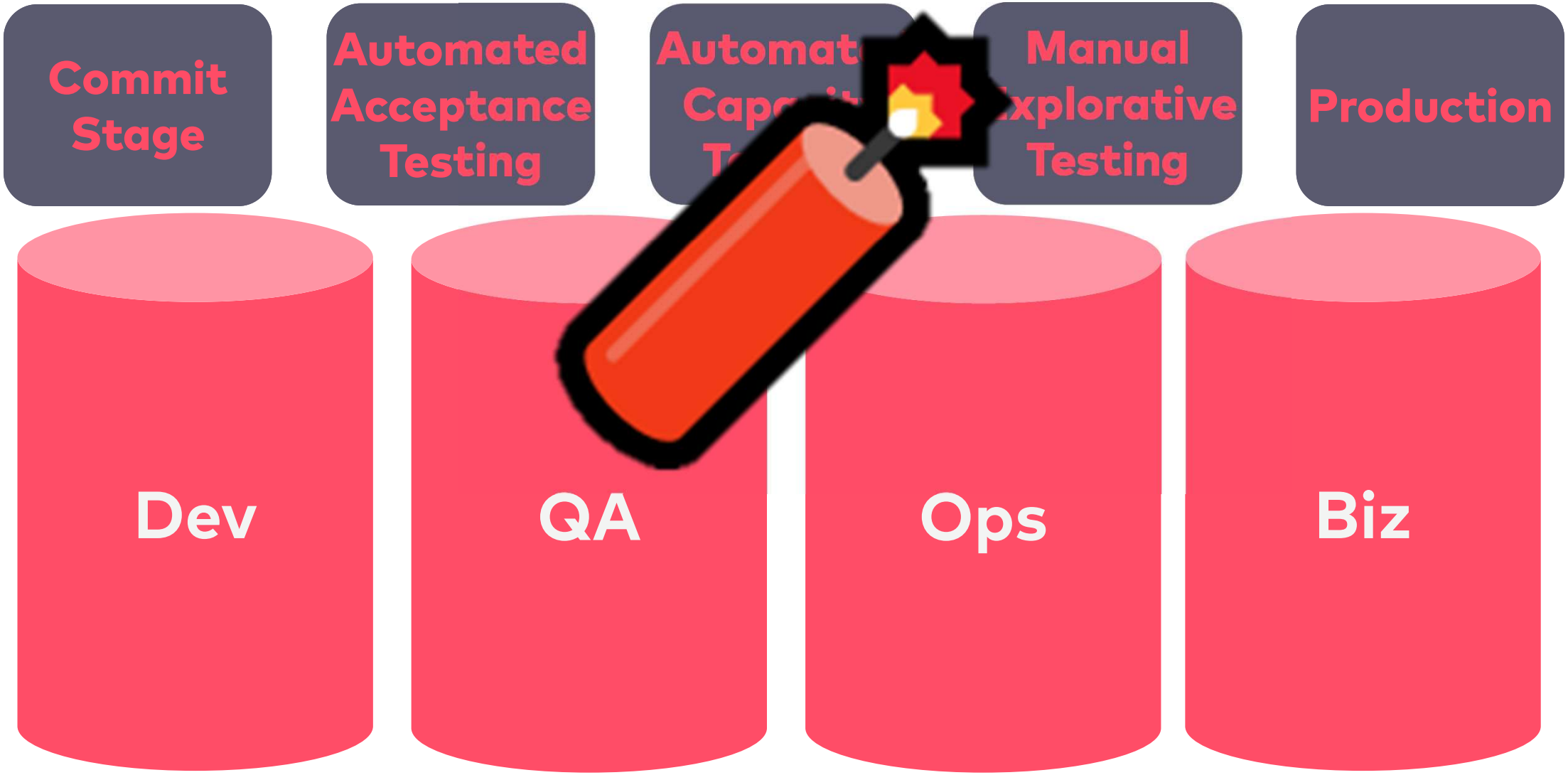
Is DevOps Enough


Commit Stage
Automated Acceptance Testing
Automated Capacity Testing
Manual Explorative Testing
Production
Dev
Ops

DevOps + CD Pipeline

- Dev: commit stage
- Ops: production
- What about tests?
- What about business feedback from production?


Commit Stage
Automated Acceptance Testing
Automated Capacity Testing
Manual Explorative Testing
Production
Dev
QA
Tests
Ops
Biz
Feedback


Commit Stage
Automated Acceptance Testing
Automat
Manual Explorative Testing
Production
Dev
QA
Ops
Biz

CD Without Collaboration?

- Limited success
- CD might also make organizational issues obvious.
- Bug or feature?

DevOps =
DevOps Engineer


DevOps =
DevOps Engineer

DevOps =
Collaboration & Culture

Conclusion

Conclusion

- Continuous delivery is a process to optimize software delivery performance
- Continuous delivery improves
  - Security
  - Time to restore service
  - Time spent on new features
  - Burnout
  - Organizational performance
- ...and also time-to-market

Conclusion

- Continuous Delivery even impact organizational performance!
- Extremly valuable!
- Needs collaboration, not a DevOps engineer