

Improve quality by monitoring and observing



Adam Toth
Quality Architect I

Why we need to improve QA processes continuously?

Early feedback

Time To Market

Digital Transformation

CI/CD

Test automation ROI

Unsecured systems

Agile Transformation

Product quality

Save Money

Client wants automation

Microservice Architecture

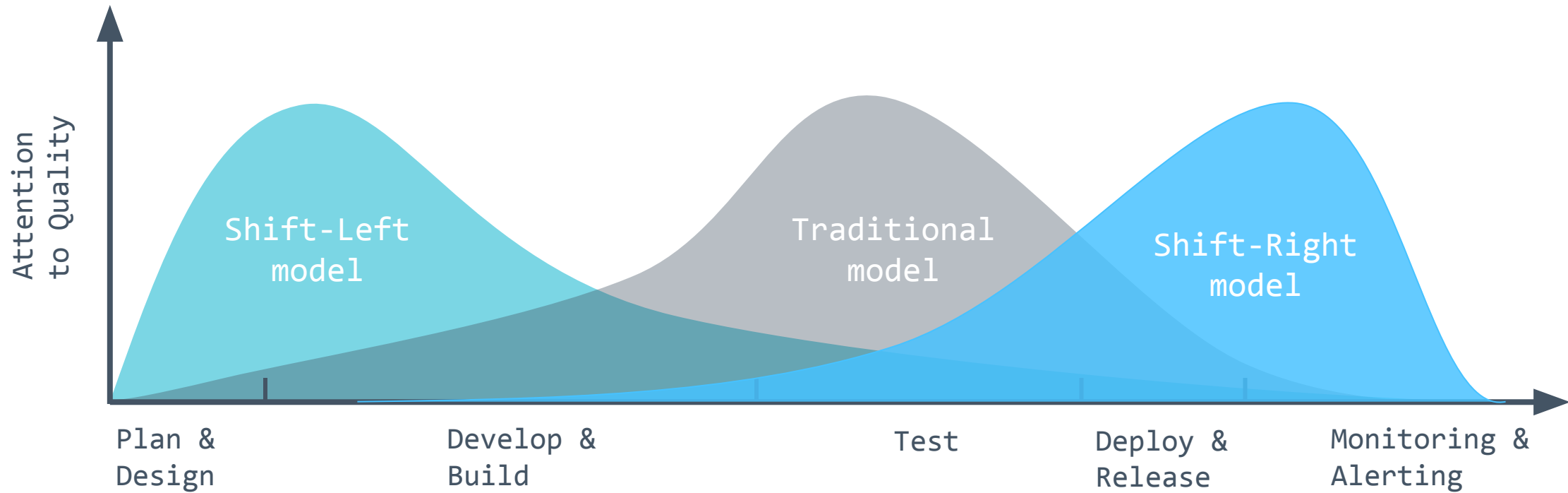
Ineffective regression tests

Poor application performance

Follow Market trends

Continuous testing

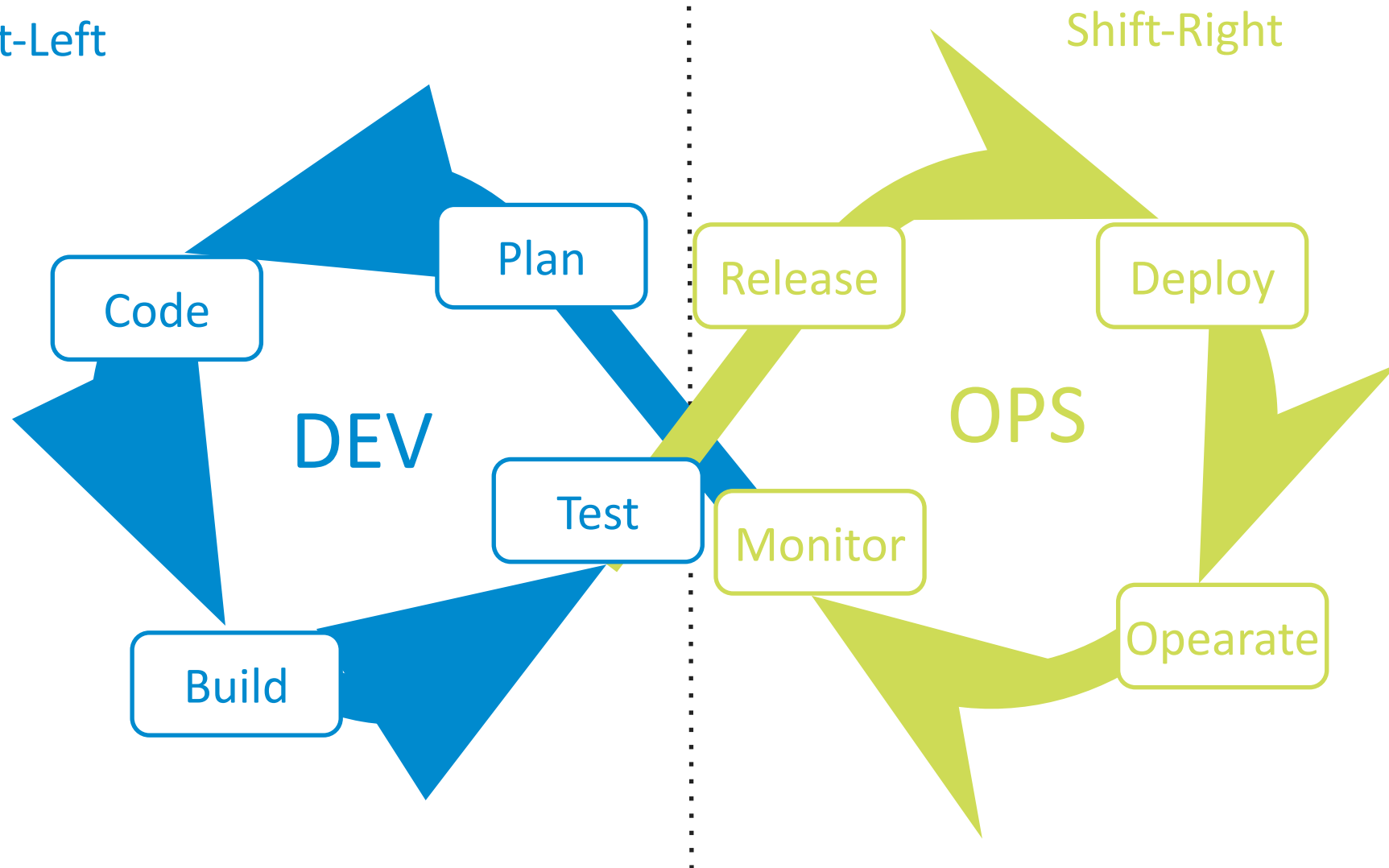
Traditional model compared to continuous testing



What is continuous testing?

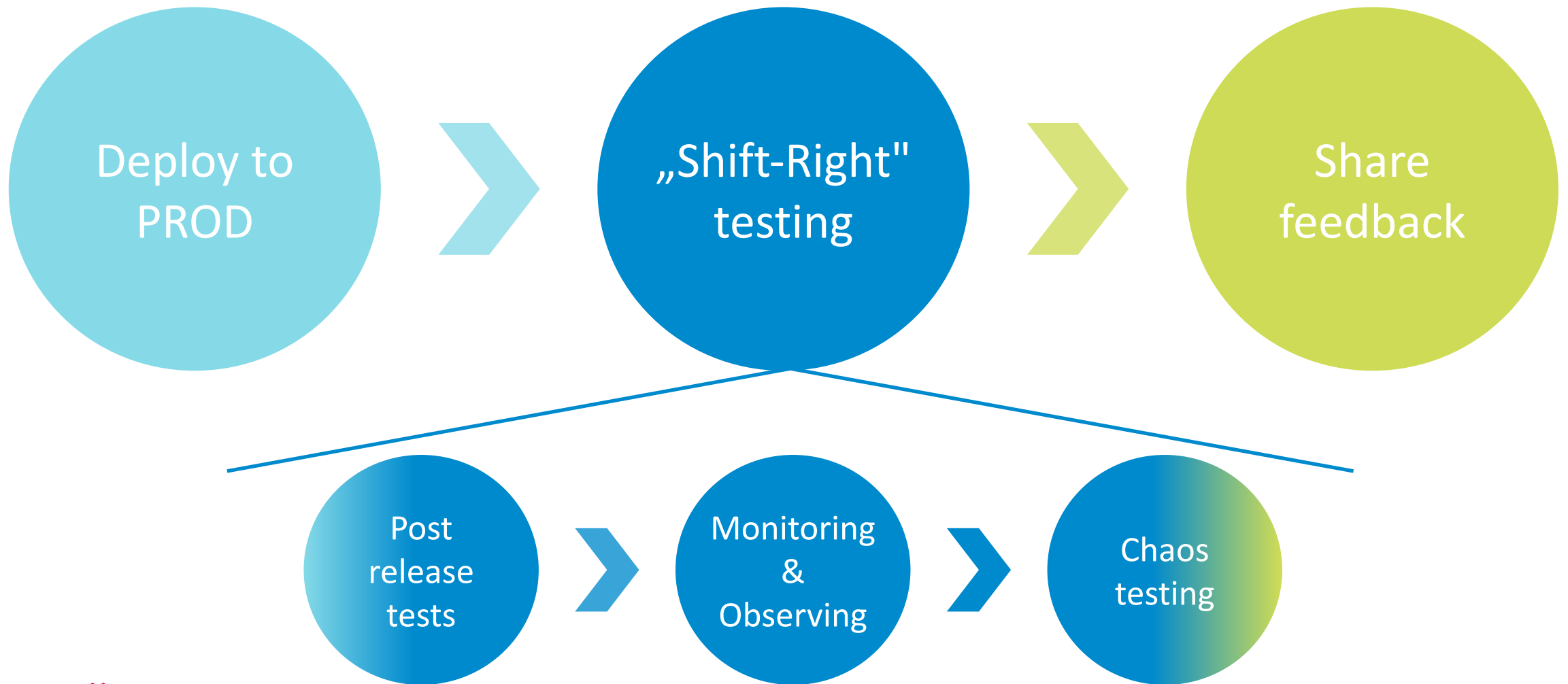
Shift-Left

Shift-Right

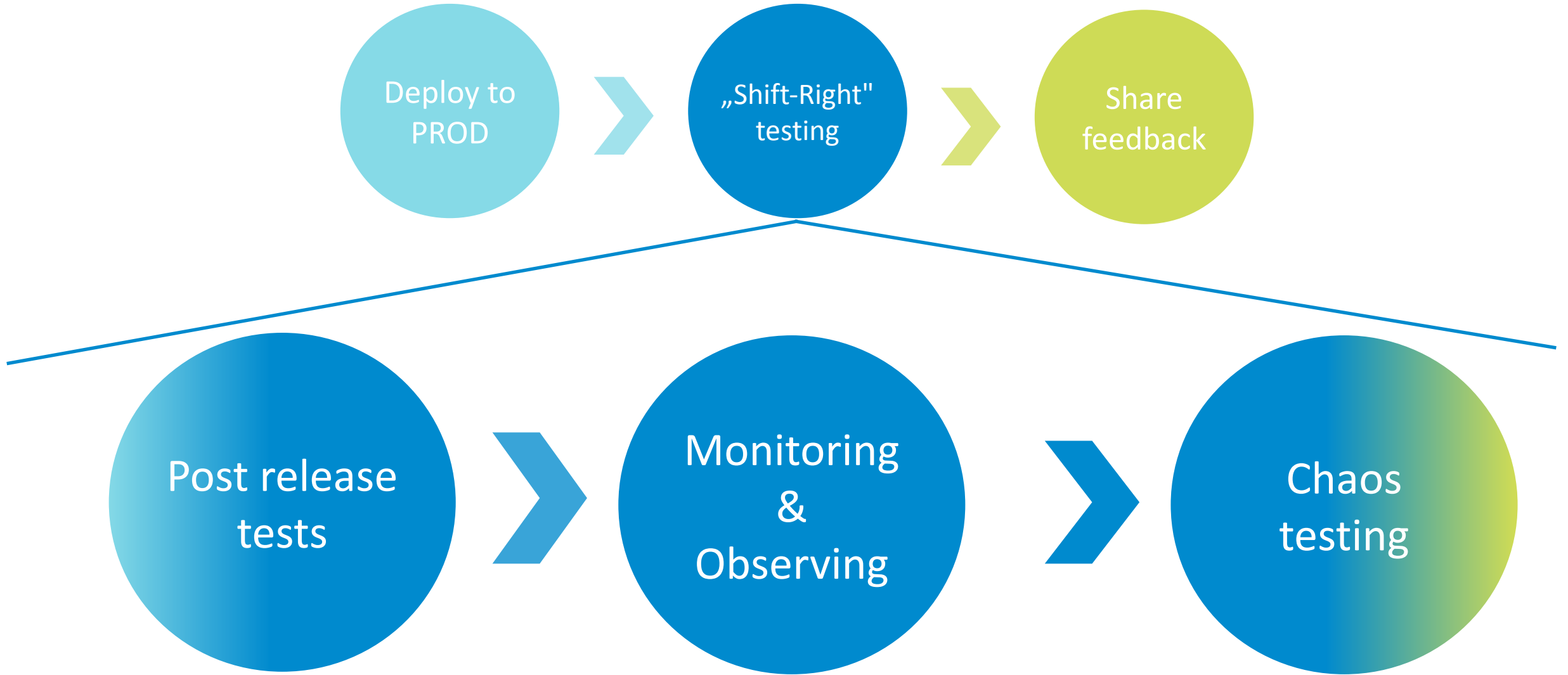


Shift-Right

How do I do this “Shift-Right”?

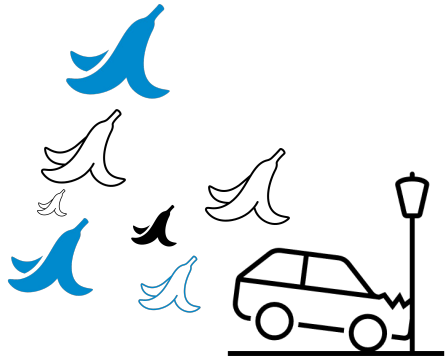


How do I do this “Shift-Right”?



Monitoring and Observing

What issues can be resolved by monitoring and observing?



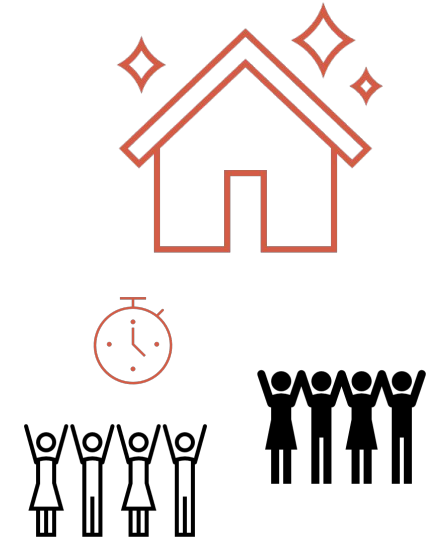
Poor product quality



Unhealthy system



Lack of knowledge about real users



Improper
Nonfunctional
requirements

What metrics do we monitor?



Infrastructure

CPU, Memory, Disk

Networking

Scaling & uptime



Application

Errors/error rate (API, Pages, DB)

Response time (API, Pages, DB)

Throughput (API, Pages, DB)



Users

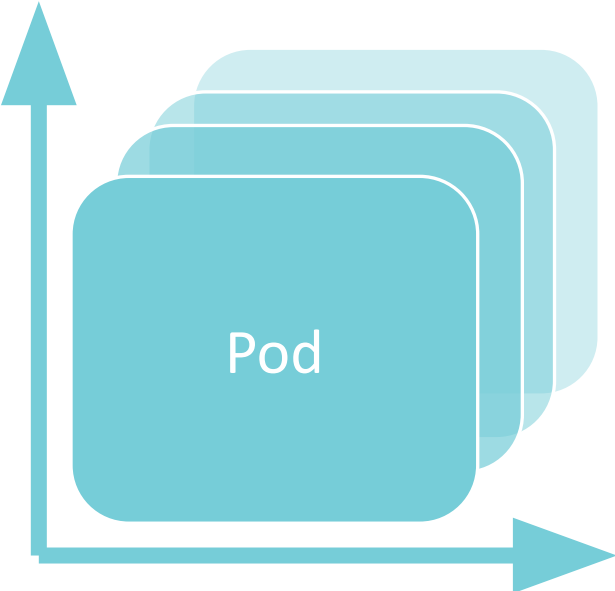
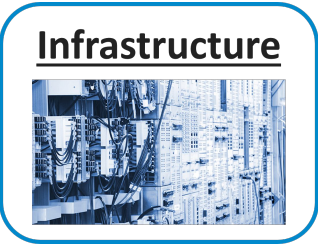
Frequent flows, dataset

Diversity of users

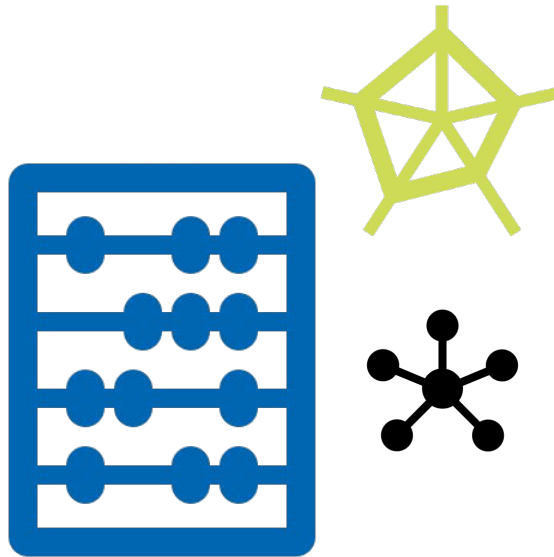
Customer tickets created/rating

Typical challenges

My challenges with Infrastructure monitoring



Increase horizontal/vertical scaling instead of application optimization



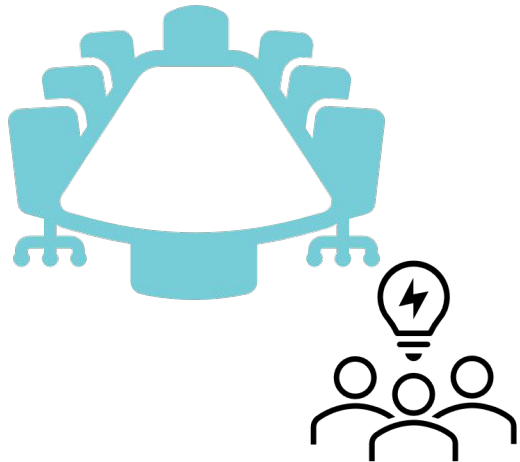
Multiple systems and monitoring solutions



Infrastructure issues cannot be transferred to Business level

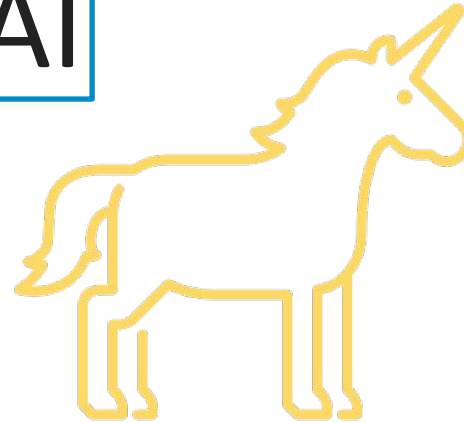
My challenges with Application monitoring

Application



Business and Dev team needs to be involved to create effective monitoring

AI



Lack of TestOps expert...and less who can share feedback effectively

`NullPointerException`

`ArrayIndexOutOfBoundsException`

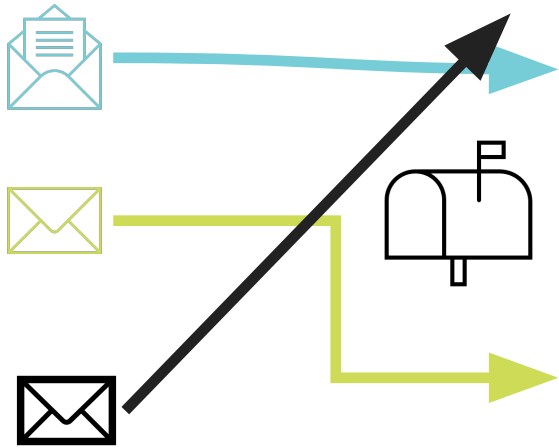
`ConstraintViolationException`

`IllegalStateException`

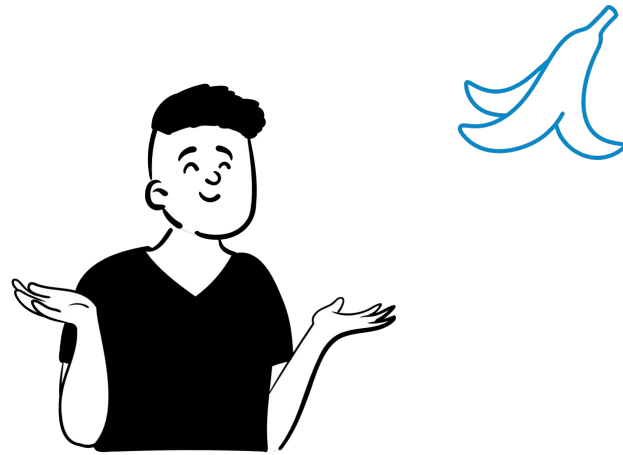
Out of ideas error

There are good patterns, but new challenges can appear anytime

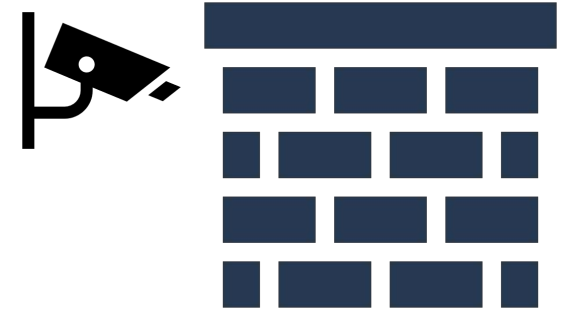
My challenges with User monitoring



Feedback is not shared with Application monitoring team



Users monitoring cause performance issues/alerts



Wrong implementation of monitoring solution

When and How?

When do I do this?



SHOULD CONSIDER FOR

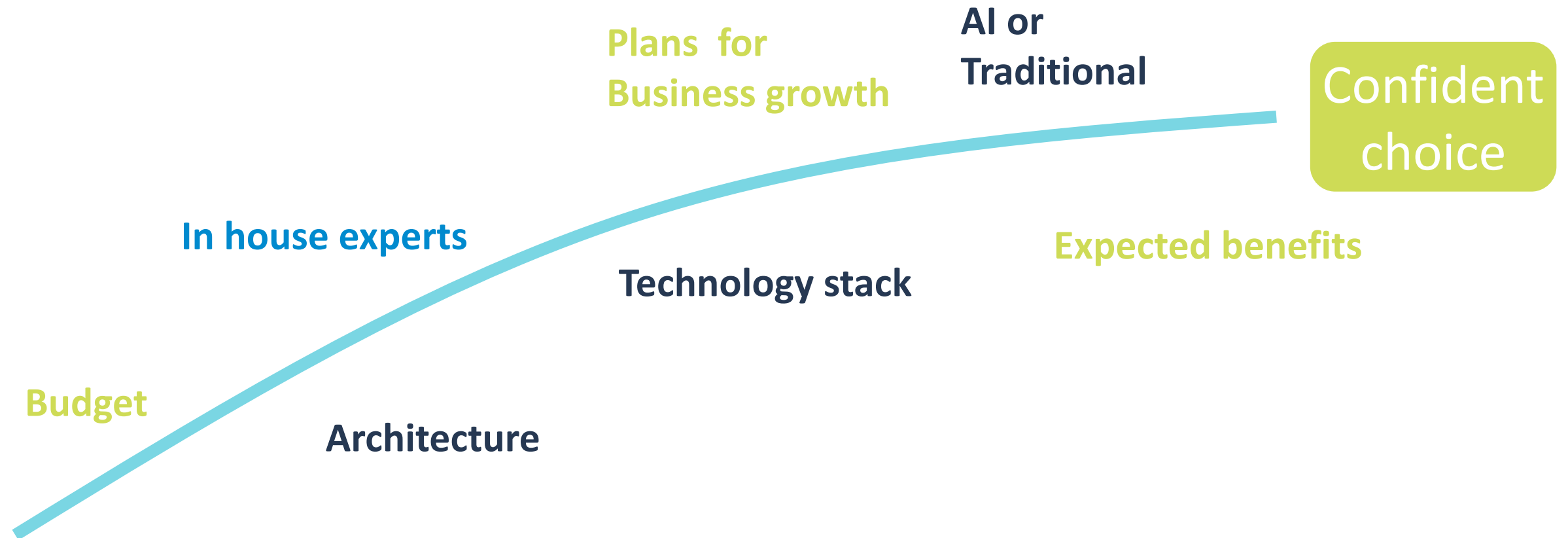
- Complex system with many users, parameters
- Quality in focus
- Proper communication channels across business and technical teams
- Long planned application lifetime
- Proper monitoring

THINK TWICE IF YOU HAVE

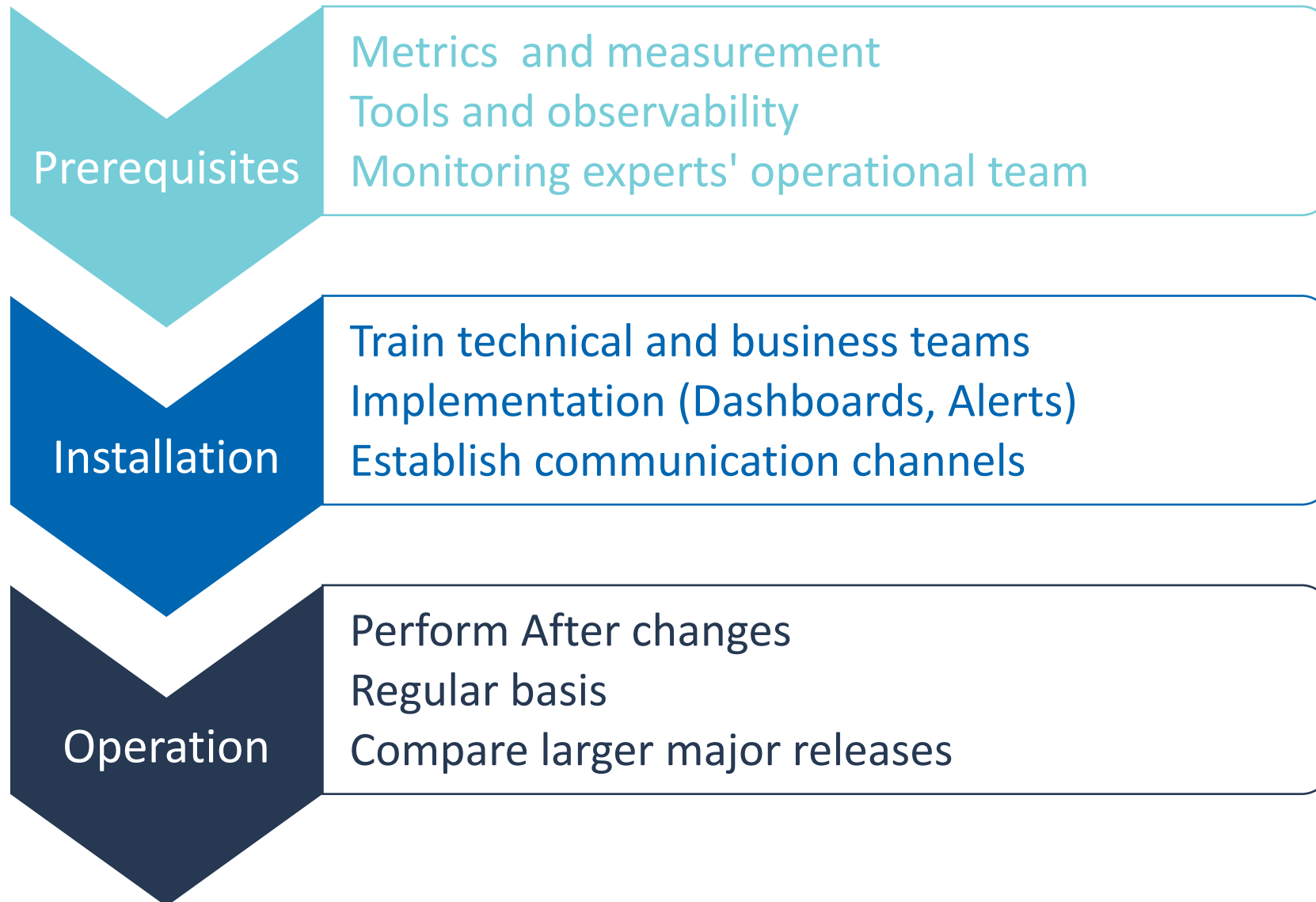
- Lack of business flows, small amount of users
- Production Feedback is not used
- Quality is not important
- Business is not involved in testing, monitoring activities
- Broken observability

Plenty of monitoring tools

What major factors need to be considered for tool selection?



How to do it properly?






Conclusion

Benefits & Challenges

Benefits

-  Short feedback loop
-  Cost effective
-  Product confidence

Challenges

-  Lack of experience
-  Observability
-  Broken feedback loop

IMPROVE QUALITY BY MONITORING AND
OBSERVING

Q&A





HUSTEF

HUNGARIAN SOFTWARE TESTING FORUM



Thank you!

Adam Toth
Quality Architect I

References

Shift-Right Testing: The Emergence of TestOps

<https://devops.com/shift-right-testing-the-emergence-of-testops/>

Continuous Testing: Shift Left & Shift Right, Get it Right

<https://www.inflectra.com/Ideas/Whitepaper/Continuous-Testing-Shift-Left-Shift-Right.aspx>

Shift left vs shift-right: A DevOps mystery solved

<https://www.dynatrace.com/news/blog/what-is-shift-left-and-what-is-shift-right/>

Continuous Performance Management Basics

https://www.splunk.com/en_us/blog/learn/continuous-performance-management.html