DevOps

een onzekere toekomst of een uitdaging?

Ivo de Bruijn & Joost van Wollingen

Go to <Kahoot URL>

- Sharing experiences of each of our companies
- Use your mobile phone to give your input

Agenda

Introduction
Context of bol.com

Journey
Moving from Waterfall to Devops

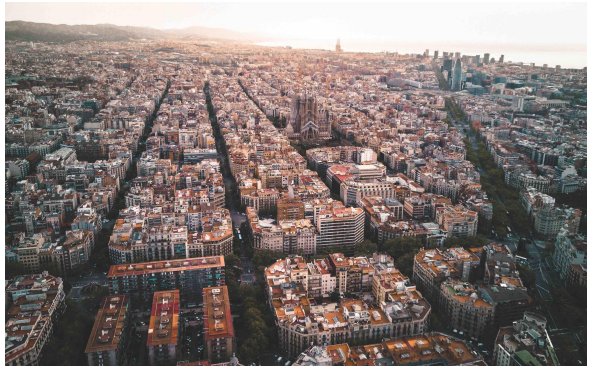
Role of the Tester
What changed for us?

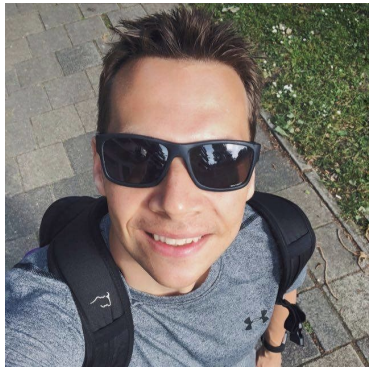
A personal story

Conclusion
Summary and Q&A

Ivo de Bruijn

Software Test Engineer, bol.com





@instabruijn

@ibruijn

Joost van Wollingen

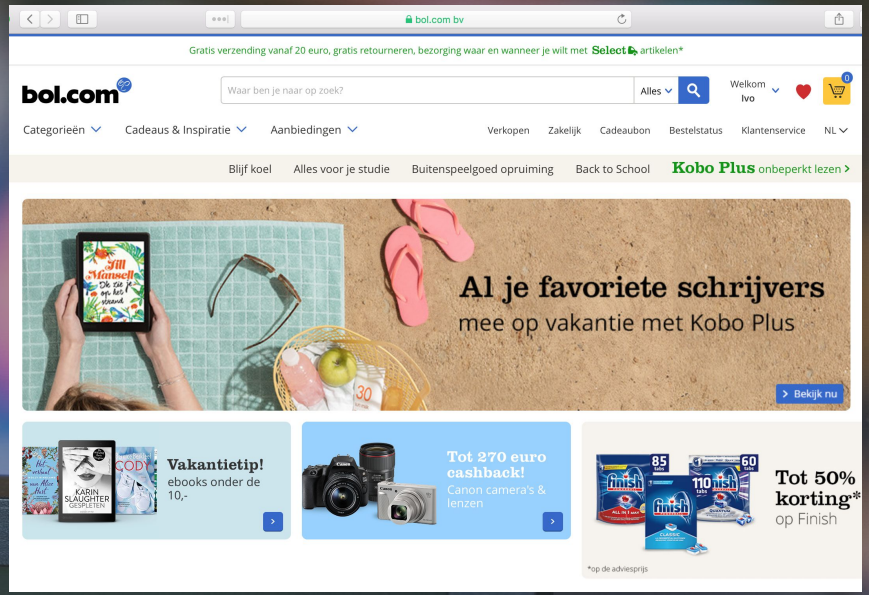
Quality & Test automation, Xebia

http://vanwollingen.nl

@jpjwolli

bol.com

de winkel van ons allemaal

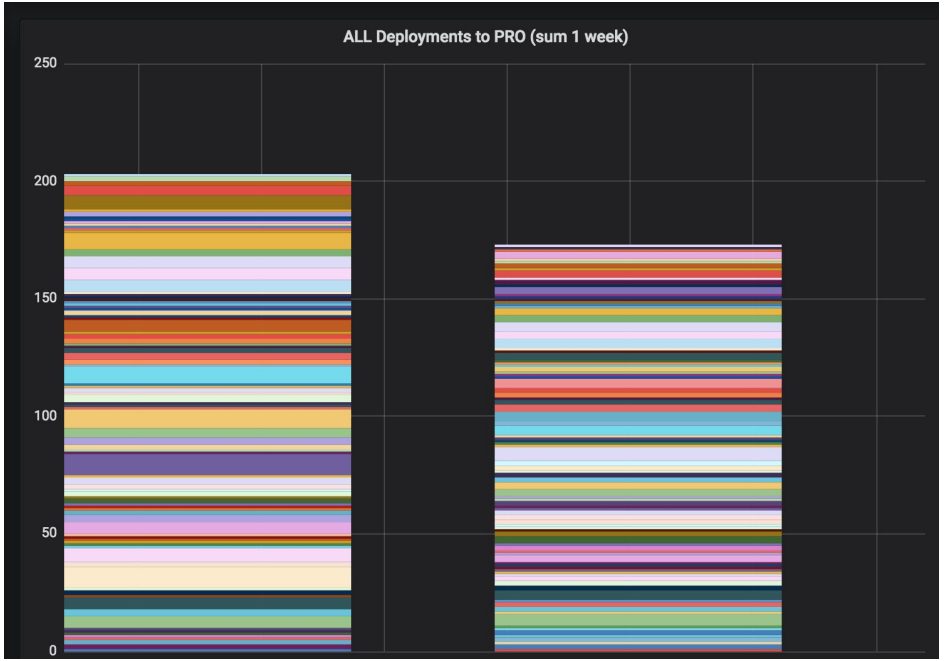


bol.com facts and figures

- 8+ million active customers
- 16+ million products
- 1400+ employees
- Location Utrecht

IT facts and figures

- 380+ people in the IT department
- 60+ Scrum teams
- 600+ microservices + legacy
- 180+ deployments per week
- In-house development



Our challenges at the time

Business and IT

Challenges for bol.com

Time to market

Rapid growth

Issues for the Business

- Unable to respond fast enough to the market
- Way too many dependencies
- Bug fixes take weeks

Issues for Scrum teams

- Test environment not available/frozen/broken
- Complete dependence on operations for deployments
- Feedback loops really long
- No insight into software running in production

Bottom line

IT is not helping the business to reach their goals


DevOps @ bol.com
Shifting both left & right

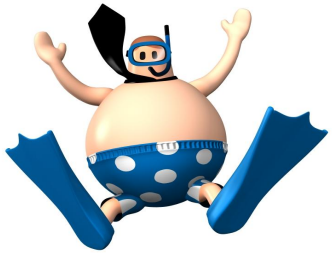
Development Process

1999

No formal process

2005

Waterfall

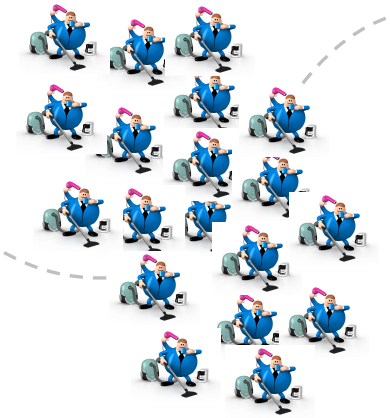


2009

Introduction Agile

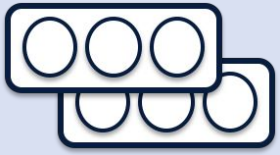
2016 - 2018

Accelerate again
Shift right & Shift left

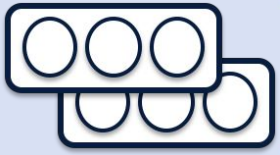


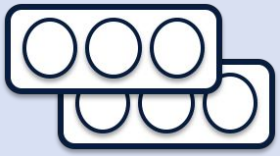
Organisation “Before”

Business

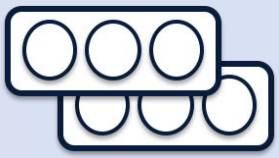


IT Development

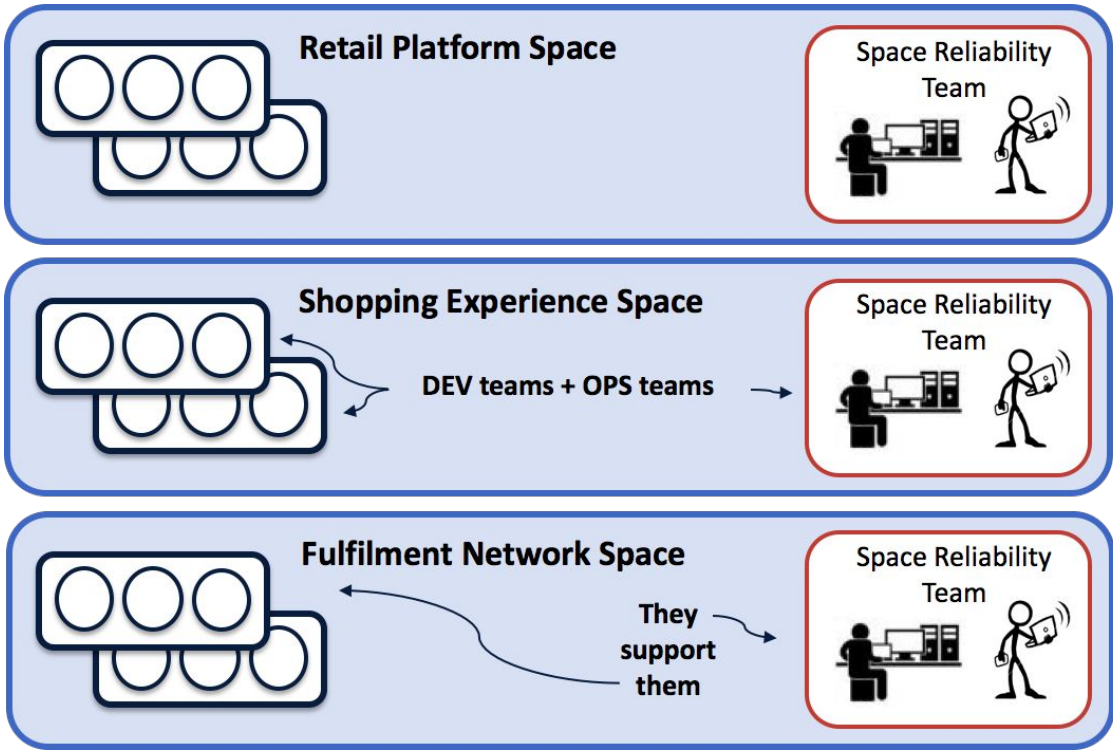




IT Operations



Organisation “After”



You build it, you run it, you love it!

I love it!

What did DevOps mean for Testers?

Knowledge from OPS to DEV

Alerting & Responsibility

Rights to push the button!

Monitoring

Metrics


Dashboard Row
Boring snapshots
100
50
0
17:30
17:40
17:50
18:00
18:10
18:20
boring snapshots Total: 1.1380 K
Error queue: boring snapshots
Segment Updates
15.2 K
ERR Q: Segment updates
Log errors
4
Log warnings
16
Log infos
77
Shortages
Reservations received
200
150
100
50
0
17:30
17:40
17:50
18:00
18:10
18:20
smartSummarize(reservation_received, "30min", "sum")
Shortages
500
400
300
200
100
0
17:30
17:40
17:50
18:00
18:10
18:20
smartSummarize(shortage_created, "30min", "sum")
Shortages
total
consolidateBy(12,"sum") 64
consolidateBy(2,"sum") 556
consolidateBy(9,"sum") 0
Reservation fulfilment
Reservations supply cancelled
1.0
Reservations made
95
Preferred Warehouse

Alerting


bol.com
Team3d PRO IT-Ops Dashboard - 0 Alerts - Refresh in 11.3 Last refresh @ 2016-10-05 21:44:26.
[hide clock] [hide soft alerts] [hide ack'ed alerts] [hide scheduled downtime] [hide muted notifications] [hide menu]
There are no alerts!

Results

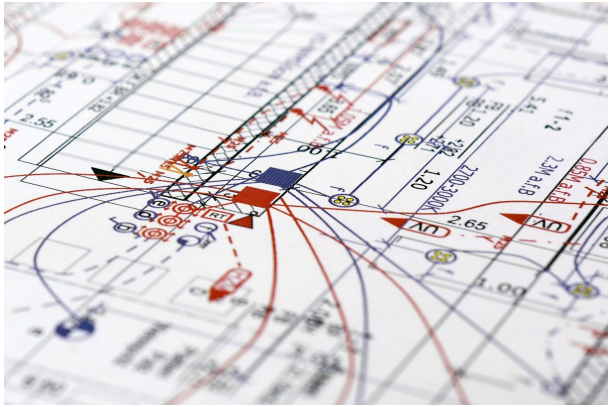
- New responsibilities encouraged adoption of DevOps
- Software Quality becomes extremely visible (and painful!)
- Teams get better feedback from PRO

Testing in isolated environments

Shift left

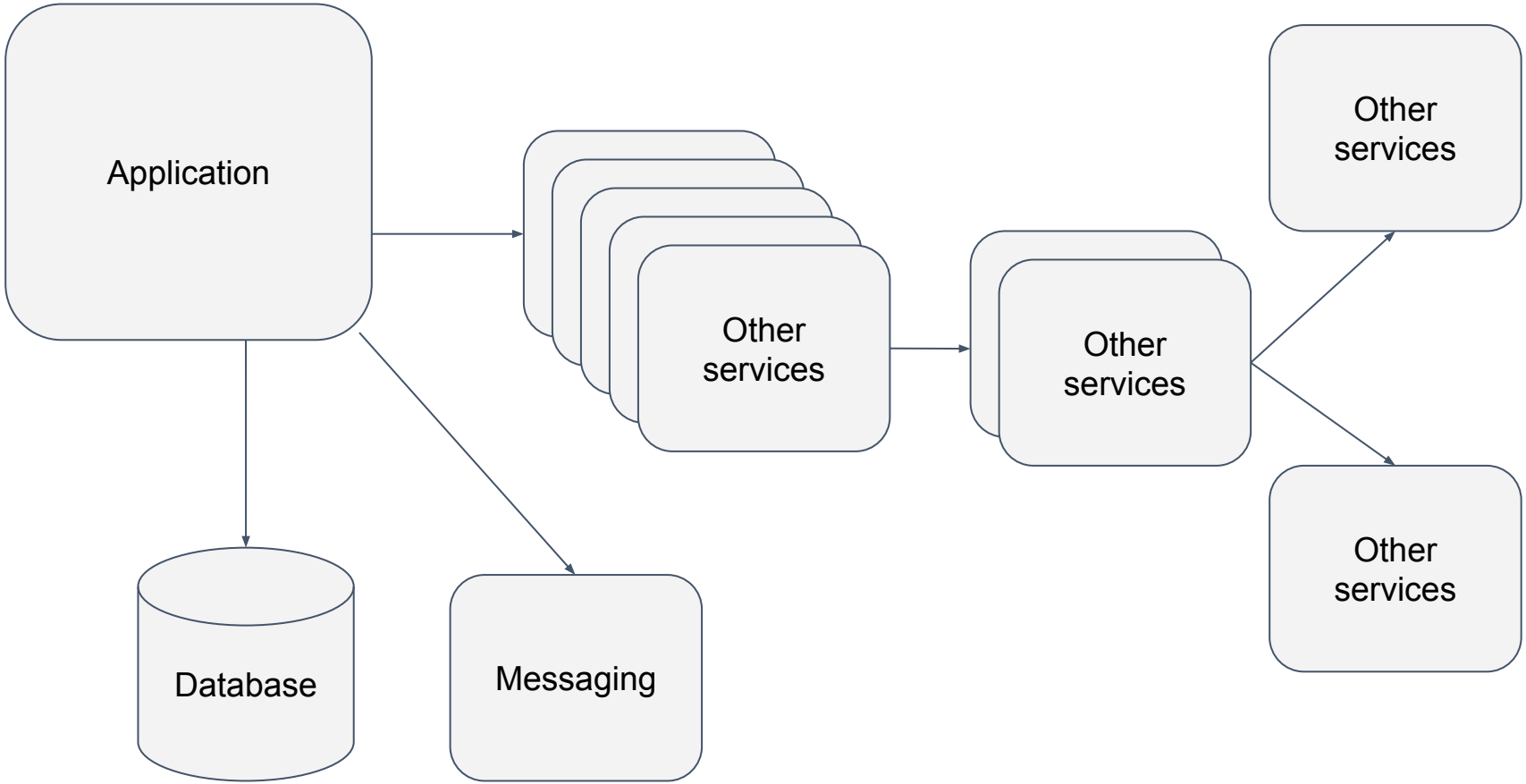
Changes with a big influence on testing

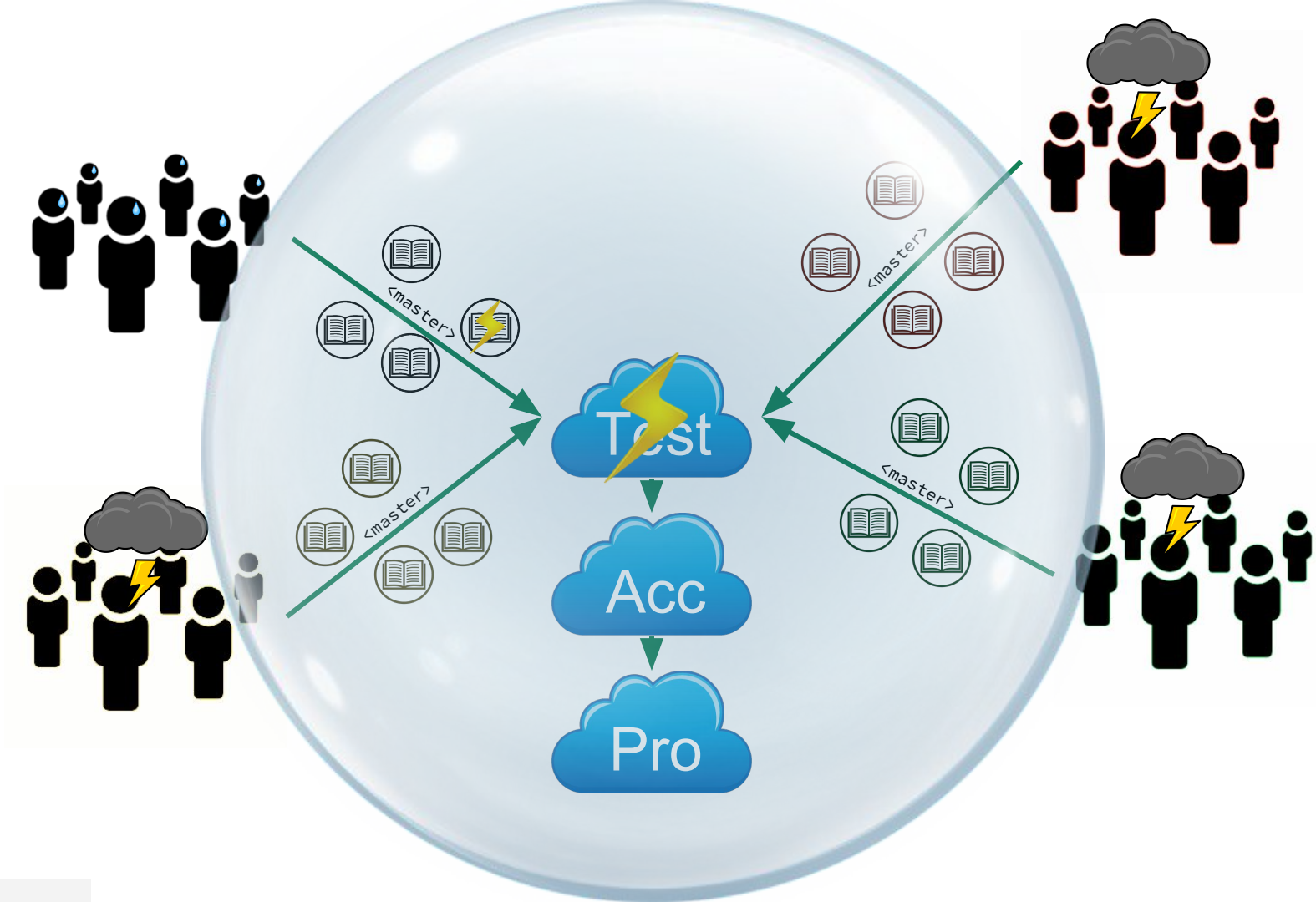
DevOps
(shift right)



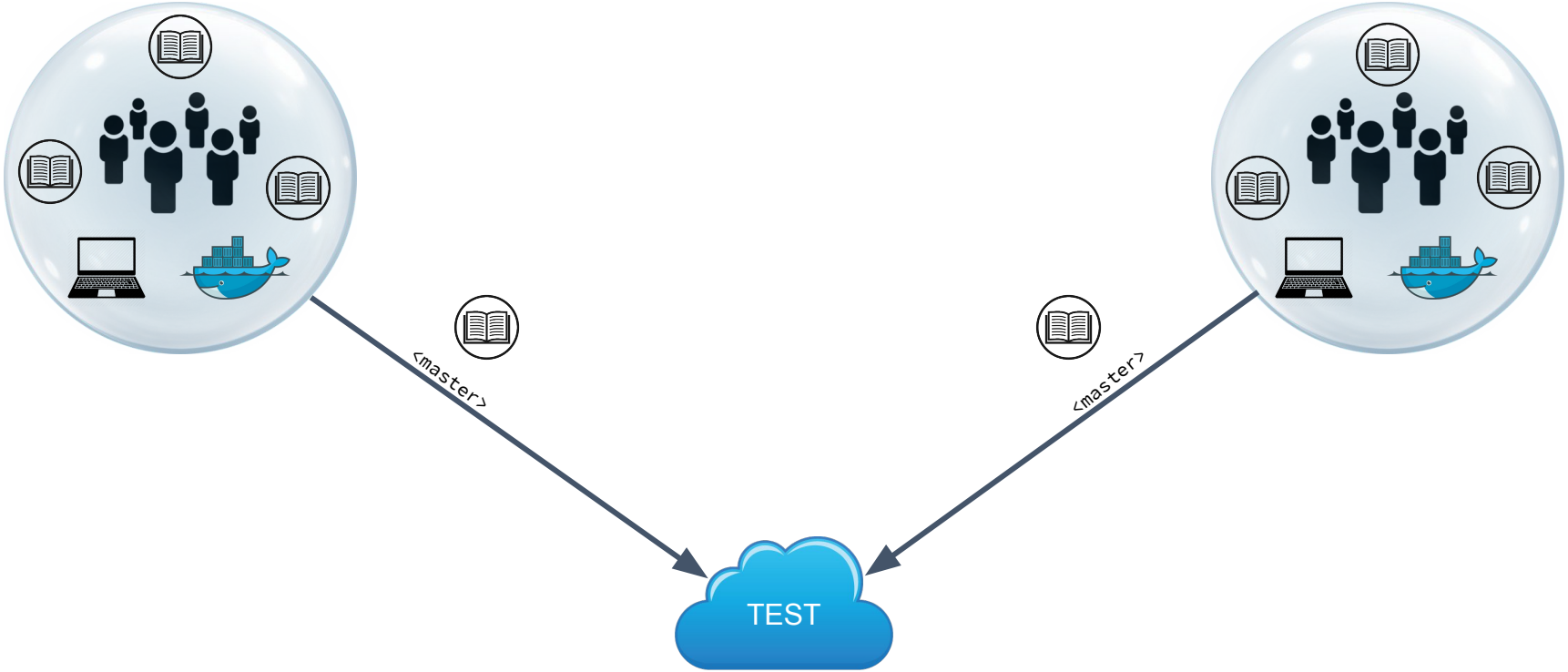
Service Oriented
Architecture (SOA)

What was the problem?

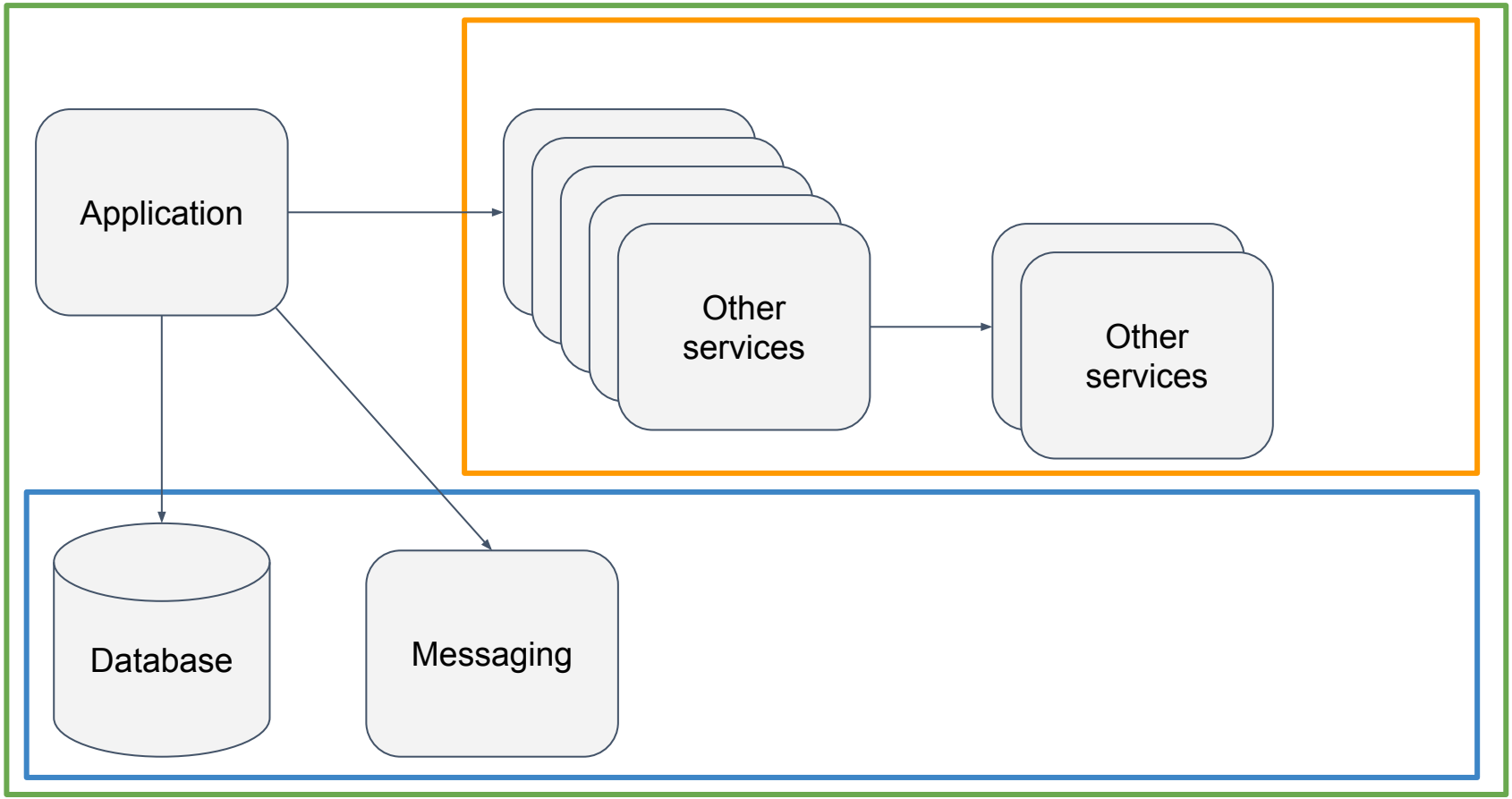



<master>
<master>
<master>
<master>
Test
Acc
Pro

Isolated environments

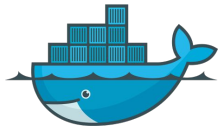


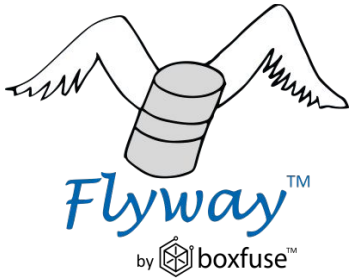
What are we isolating?



Tools

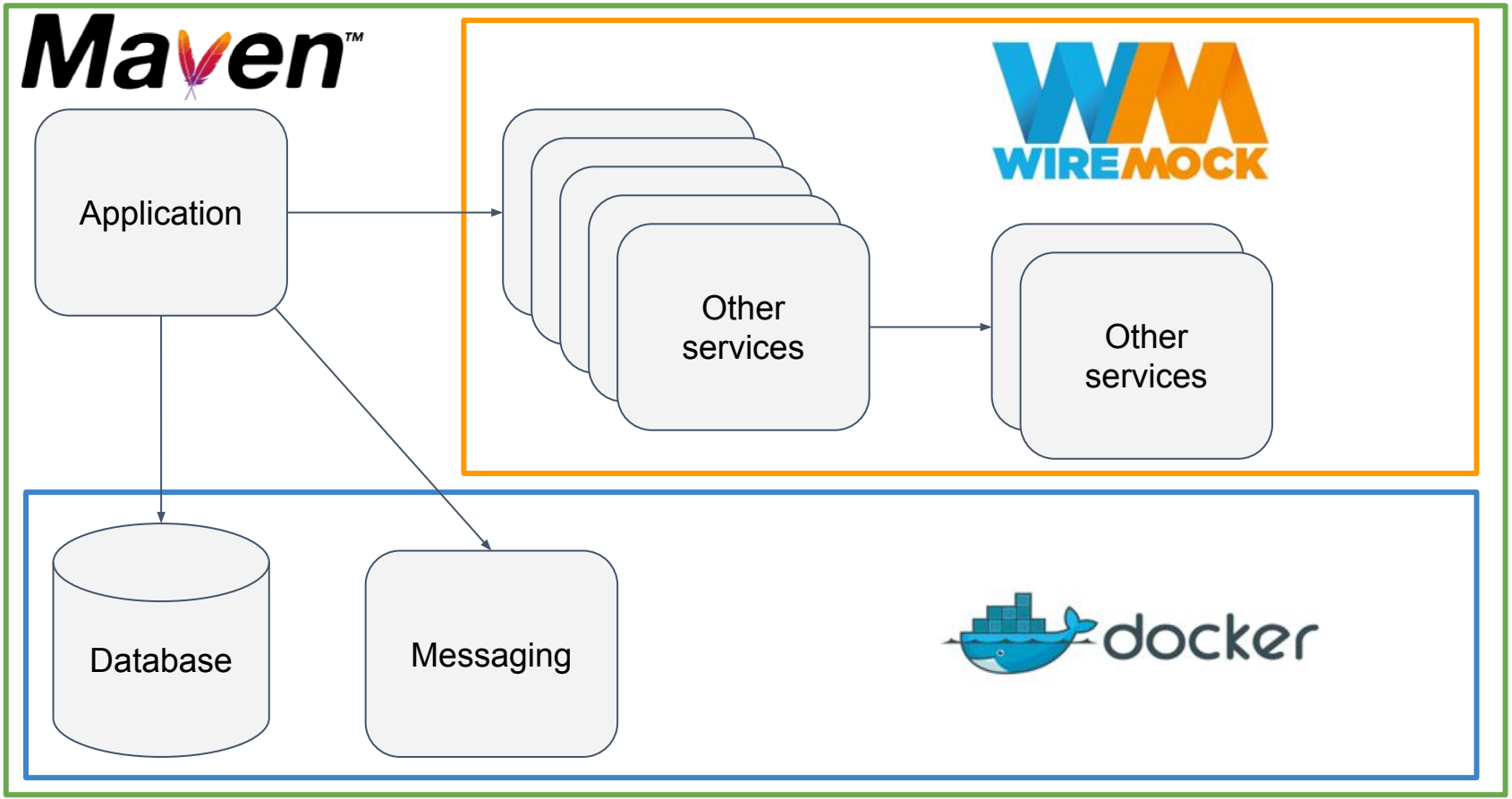
- Docker
- Database migrations
- Maven docker:plugin
- Wiremock





Maven™

Create your isolated environment



Benefits of testing in isolated environments

- Speed up your feedback cycle from days to minutes
- Isolate yourself from others and isolate others from you
- Taking full control of your own test environment and test data
- Get rid of the “works on my machine” syndrome
- Improvement of the quality of the software you deliver

Other benefits for testing

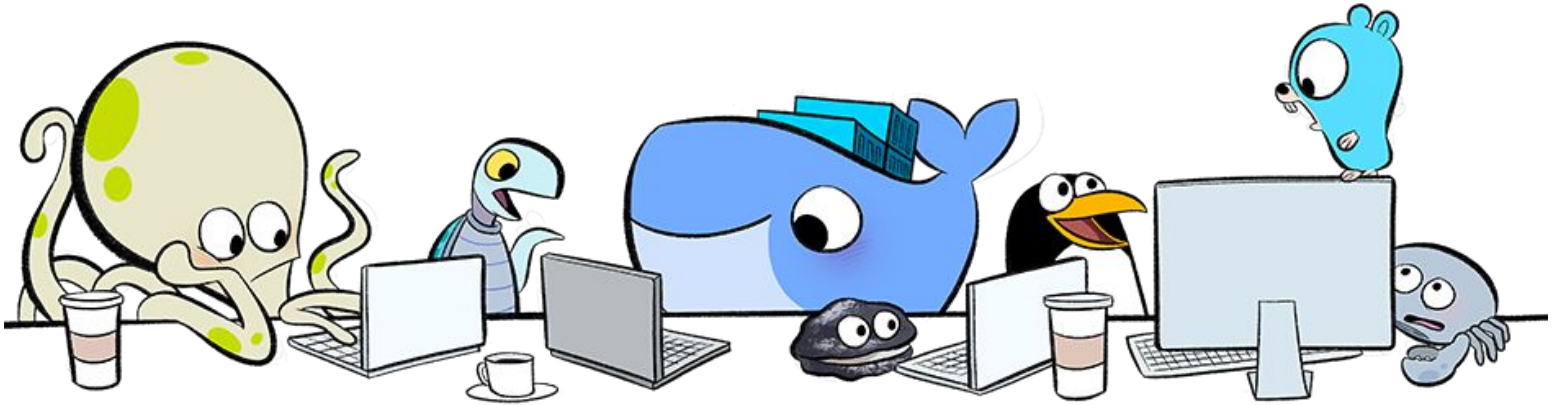
- Testing as a team responsibility (same tools)
- Testing is part of the user story
- Learn more about your application
- Testing the resilience of your application

Some lessons learned

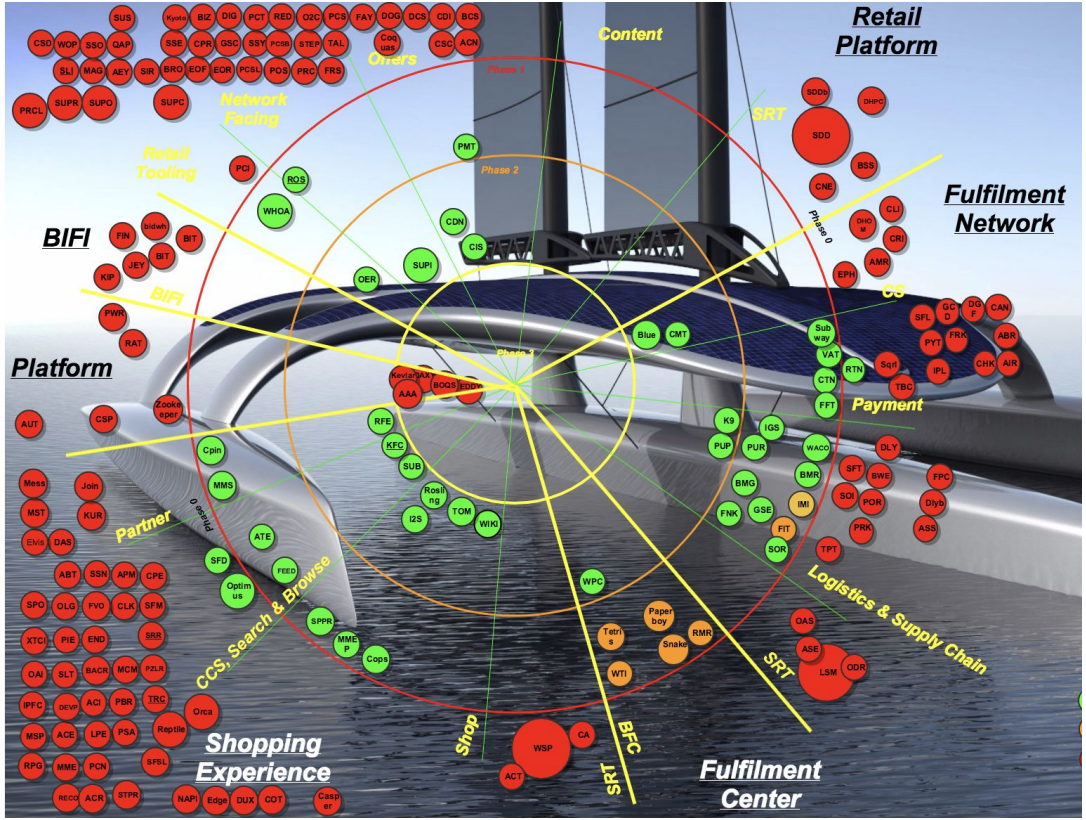
- Run tests in parallel (as much as possible)
- Really isolate your tests
- Make integration tests part of the build
- **Just Do It! Then learn and adapt**

Training: docker-for-testers

“Introduction training on Docker and how to create isolated environments.”



Maturity program



Test Autonomy

Moon to Mars

when environments are:	then tests become:
Isolated	Providers of fast feedback
Fast to create and destroy	Easier to maintain
Highly controlled	Reliable
Production-like	Non-intrusive

Phase 1: Isolation

Docker | Stubs | Fast feedback

Phase 2: Integration

Int. / Build: Exploratory, End-to-end, Component, Integration, Unit

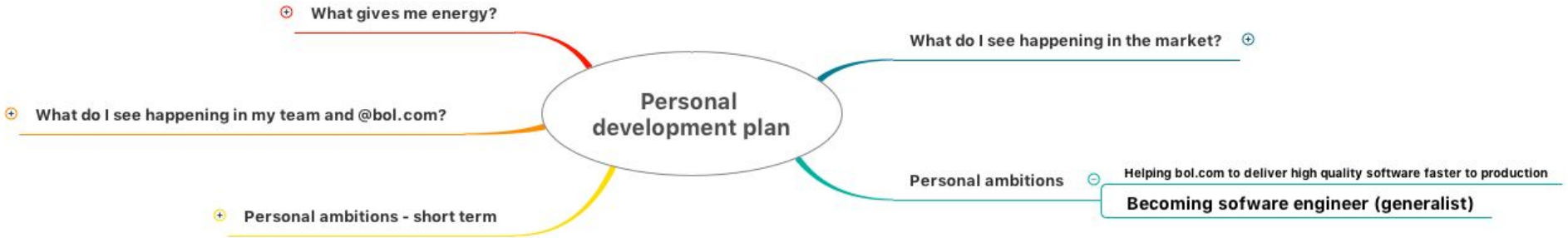
Test the right risks in the right place | Ownership | Proper test data

Start your journey today!

Test engineer to SW engineer

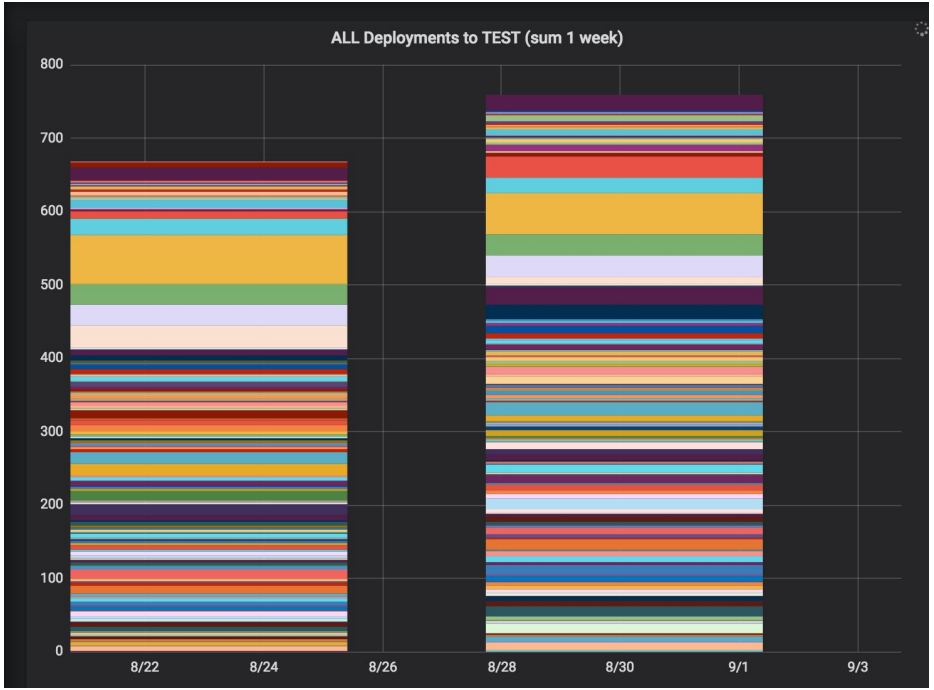
My personal journey so far

Personal plan



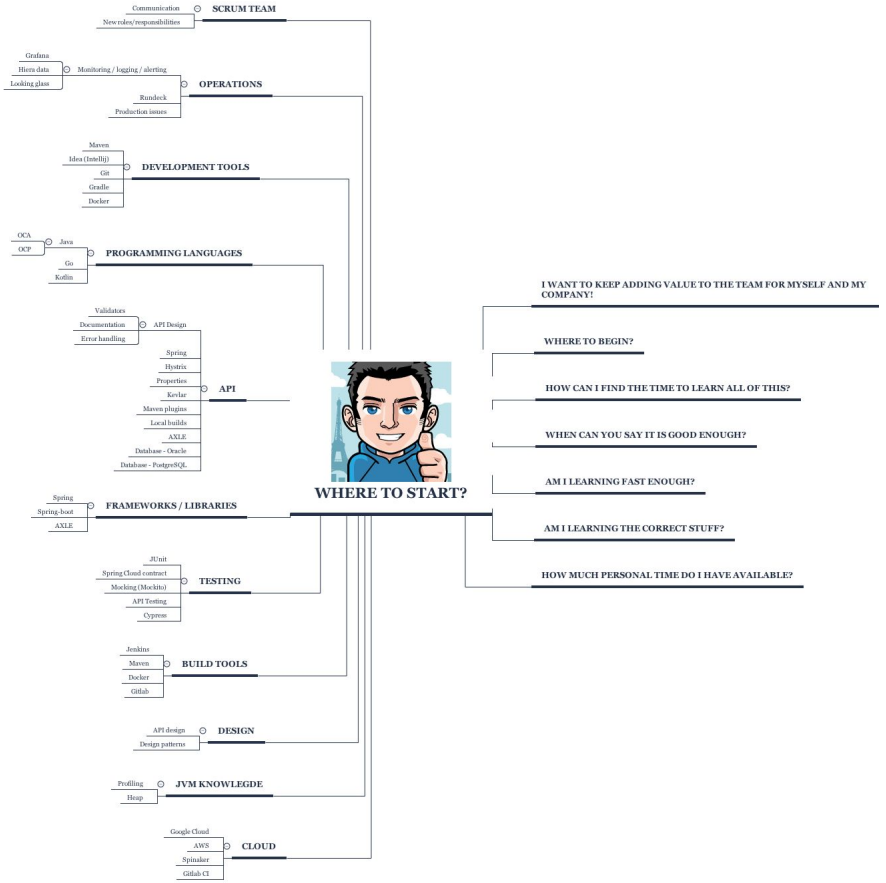
Why did I make this change?

- 180+ deployments per week PRO
- 700+ deployments per week TEST
- I wanted the technical challenge
- Learn to write application code



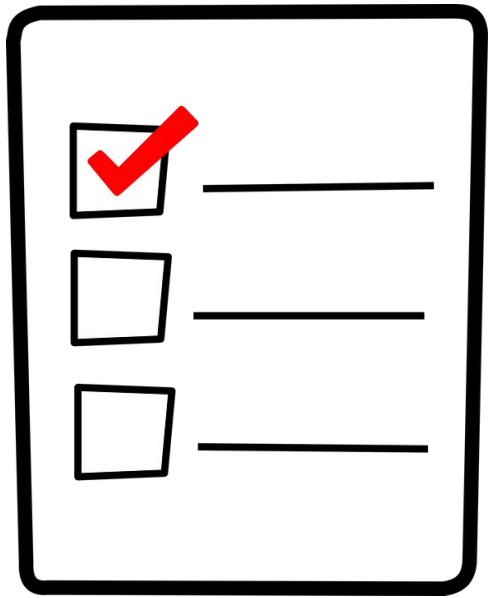
Where to start?

- A lot of questions
- Some doubts
- Really a lot to learn

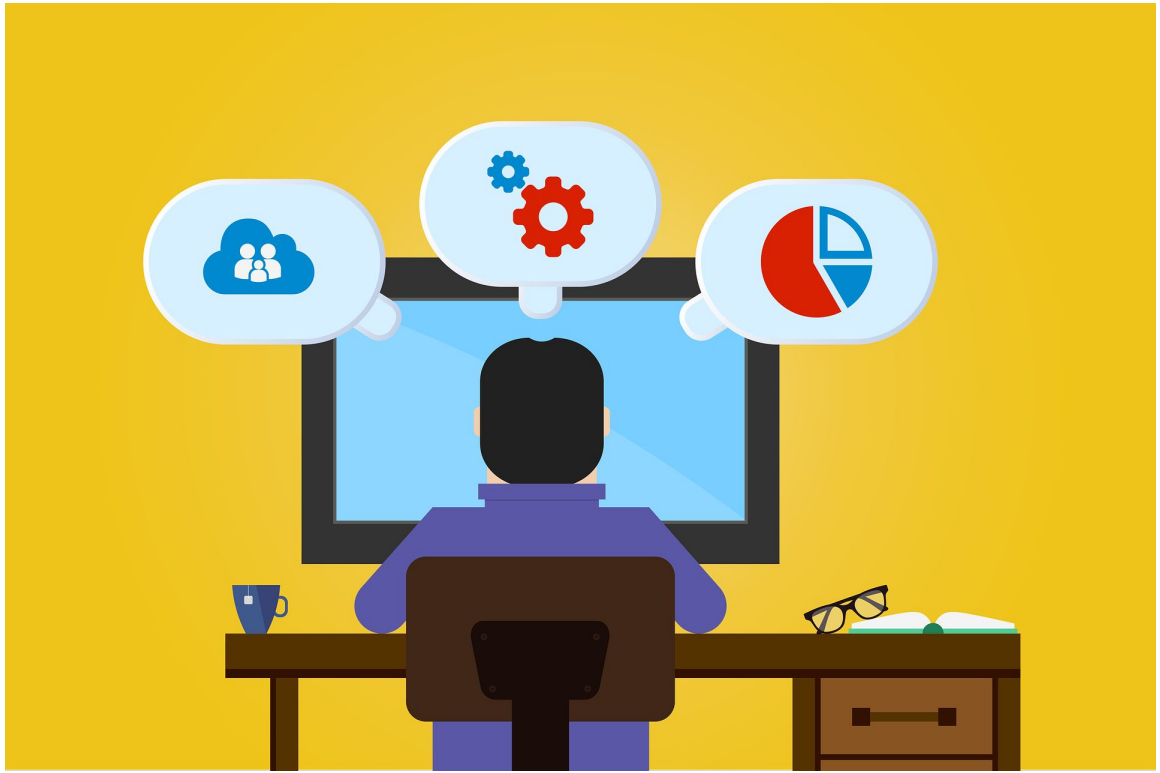


Making a list

1. Basic Java Oracle Certified Associate Java (OCA)
2. Basic Kotlin
3. Basic Maven
4. Build tools (Gitlab CI, Jenkins)
5. Basic Spring (Boot)
6. Oracle/PG database integration JDBC
7. Testing frameworks (Mockito, Spring Cloud Contract, Spring test)
8. ...



Planning (long & short term)

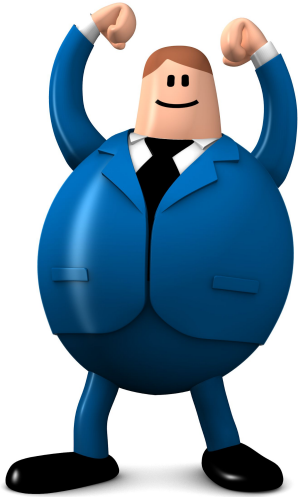


On track?

Measure

Challenge

Support



bol.com

Colleagues

Coach


What’s happening elsewhere?

“… the software testing field is in the midst of a period of extreme change.”

-- The Future of Software Testing
(a report by QASymphony)

Let’s get one thing clear...

AntonyMarcano
@AntonyMarcano

Replying to @lanettecream and 2 others

If you've been writing test code, you are a developer. You just happen to specialise in the domain of testing, just like some developers specialise in the domain of finance.

8:20 PM - Sep 1, 2018

3 See AntonyMarcano's other Tweets

Evaluating software value

Alan Page
@alanpage

Replying to @ard_kramer and 2 others

No, the only person able to evaluate the value of software is the end-user. A person playing the role of a tester can be a proxy, but it's only a guess.

3:24 PM - Aug 30, 2018

3 See Alan Page's other Tweets

“TEs need to be involved as engineers to remain first-class citizens.

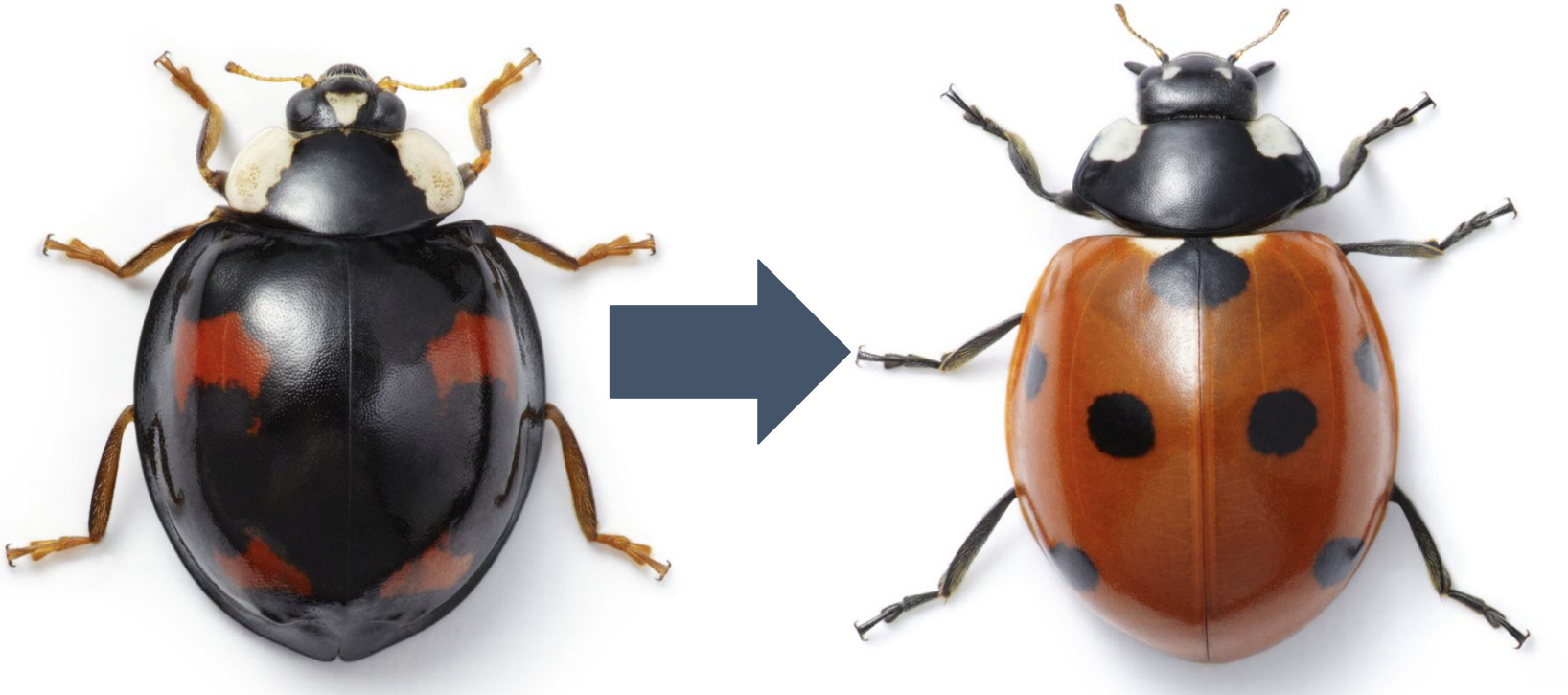
…a mix of technical skills that developers respect and a user facing focus…”

-- How Google Tests Software

Modern Testing Principles

1. Our priority is improving the business.
2. We accelerate the team, and use models like Lean Thinking and the Theory of Constraints to help identify, prioritize and mitigate bottlenecks from the system.
3. We are a force for continuous improvement, helping the team adapt and optimize in order to succeed, rather than providing a safety net to catch failures.
4. We care deeply about the quality culture of our team, and we coach, lead, and nurture the team towards a more mature quality culture.
5. We believe that the customer is the only one capable to judge and evaluate the quality of our product
6. We use data extensively to deeply understand customer usage and then close the gaps between product hypotheses and business impact.
7. We expand testing abilities and knowhow across the team; understanding that this may reduce (or eliminate) the need for a dedicated testing specialist.

Transformation of testing role



Information Provider

Quality Accelerant

Time for testing

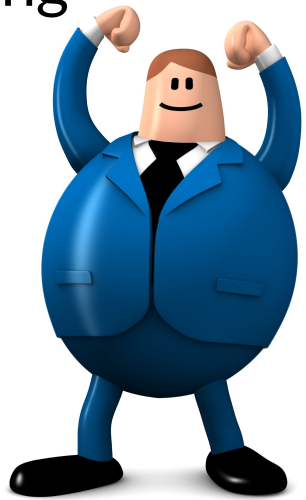
Time between releases	Time available for testing
Months to Weeks	Several days of manual testing to find bugs
Weeks to Days	
Days to Hours	
Every commit	Absolute reliance on fully automated pipelines

Mark Hrynczak, Do Less Testing, Atlassian

Conclusion

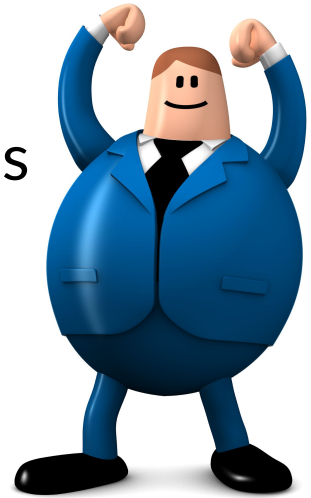
Impact on testing

- More testing in isolation
- More upskilling technical skills of a test engineer
- More “testing in production” to cover risks
- More SWE are taking responsibility for testing
- More the customer is the only one suitable to judge the quality
- More data driven decision making
- More...



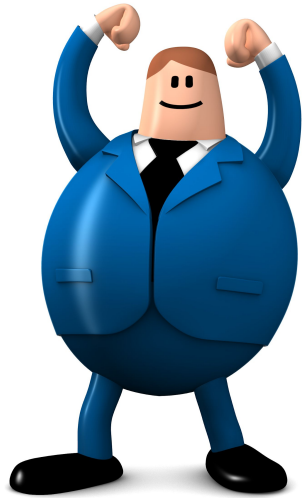
Impact on testing

- Less exhaustive up-front testing
- Less slow end-to-end testing
- Less using tools that are unfit for purpose
- Less gatekeeping by testers
- Less dedicated test specialists on agile teams
- Less...



Learnings

- Massive improvement in our time to market
- Testing becomes a shared team responsibility
- It’s all about mindset, not tools
- Developing software with autonomy, adding more value, while having more fun!




Time for questions


Bedankt!
Ivo de Bruijn & Joost van Wollingen