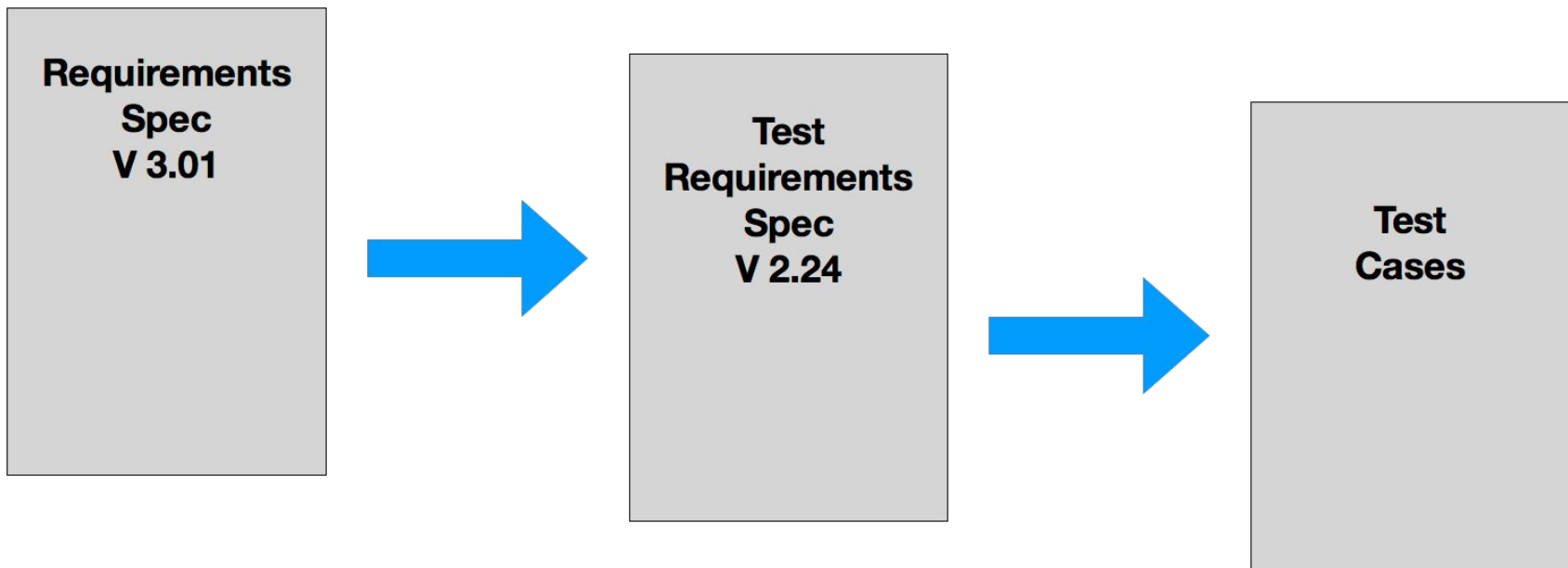# Testing Observability

Amy Phillips

Amy Phillips
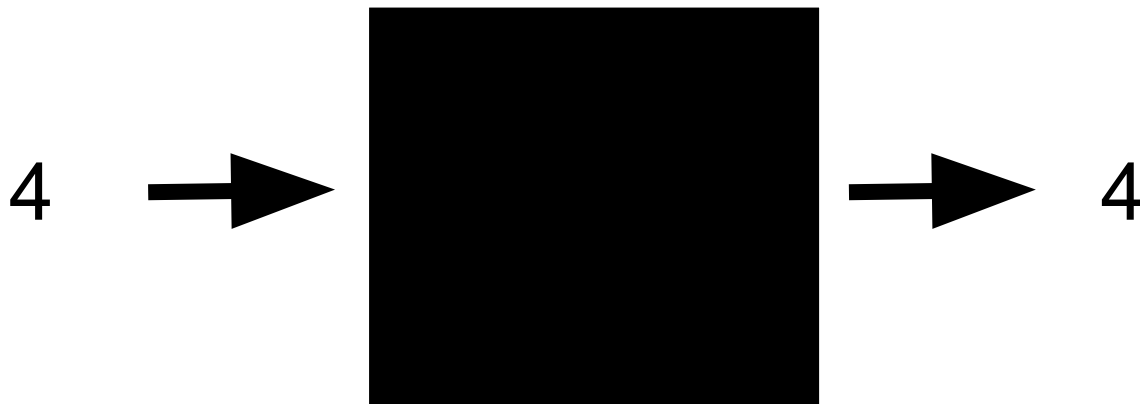
- Engineering Manager at Moo

- Previously manager and tester at Songkick, The Guardian, and Yahoo!
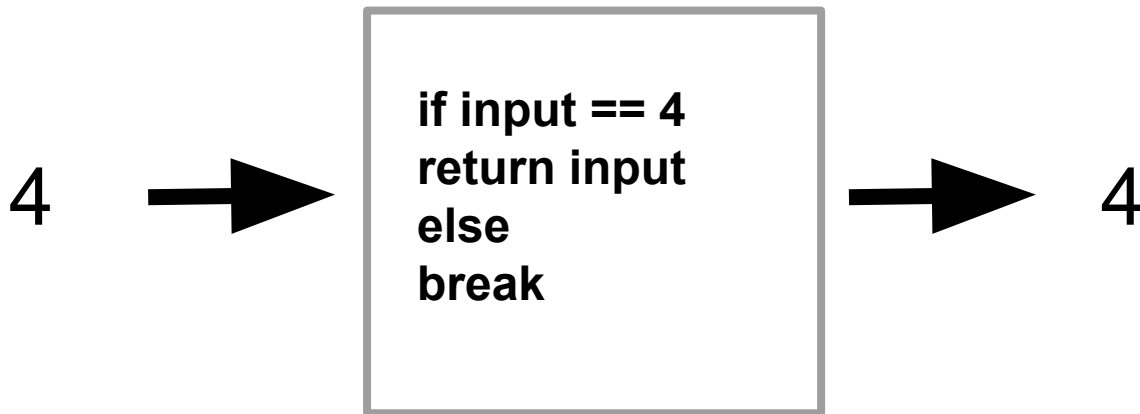
- Can be found @amyjph

The early days

Requirements Spec V 3.01 → Test Requirements Spec V 2.24 → Test Cases

Black box testing

Test case	Expected result	Test result
User saves a value of 4. Retrieve value.	System returns a value of 4	Pass

4 ➔ ⬛ ➔ 4

White box testing

Test case	Expected result	Test result
User saves a value of 4. Retrieve value.	System returns a value of 4	Pass

4 →

**if input == 4
return input
else
break**

→ 4

Releases

Build new
website

| | | | Coding | | | | | | | | |

| | | | | | Testing | | | | | | |

| | | | | | | Bug fixing | | | | | |

| | | | | | | | Retesting | | | | |

| | | | | | | | | UAT | | | |

| | | | | | | | | | Bug fixing | | |
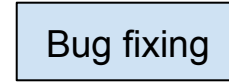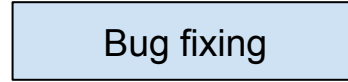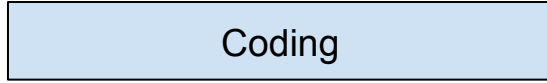
Go Live!

| | | | | | | | | | Retesting | | |

| Jan | Feb | Mar | Apr | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec |

Introducing agile

Feature 1 | Coding | Testing

Feature 2 | Coding | Testing

Feature 3 | Coding | Testing

Integration testing

Bug fixing

Go Live!

Mon Tue Wed Thur Fri Mon Tue Wed Thur Fri Mon

Fluid testing

Acceptance testing

Manual

Automated

Pairing

Code reviews

Error log reviews

Ad Hoc tools
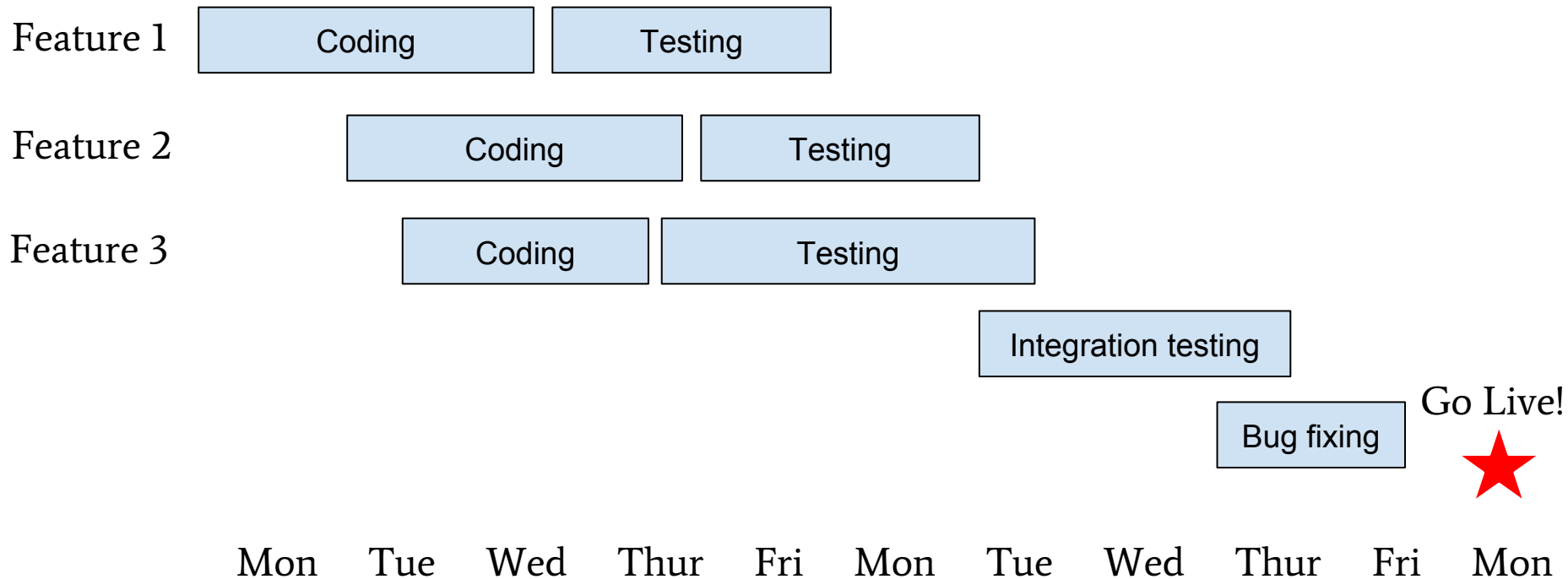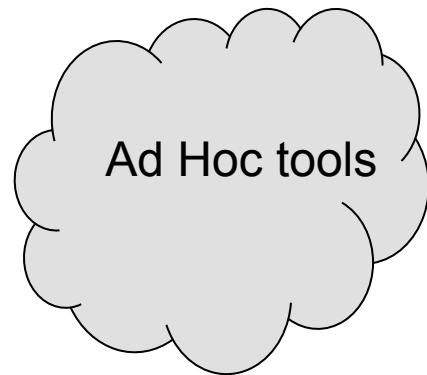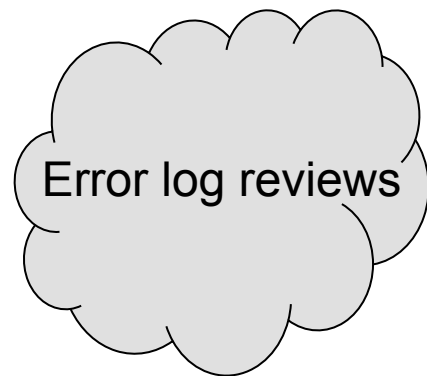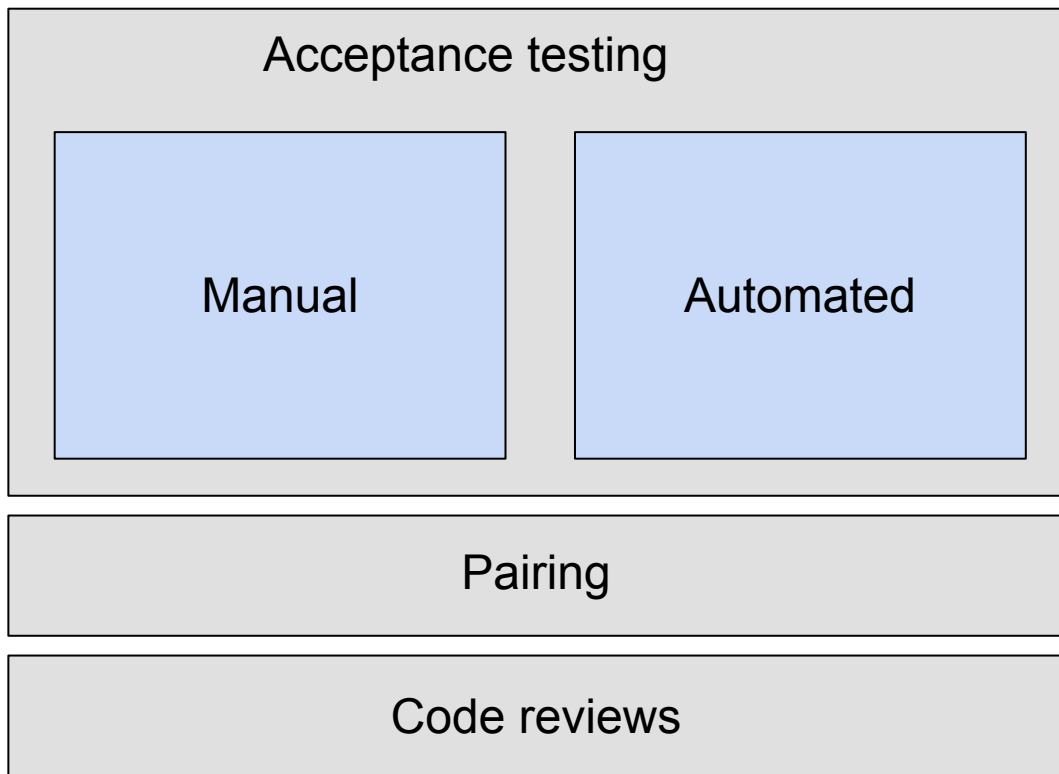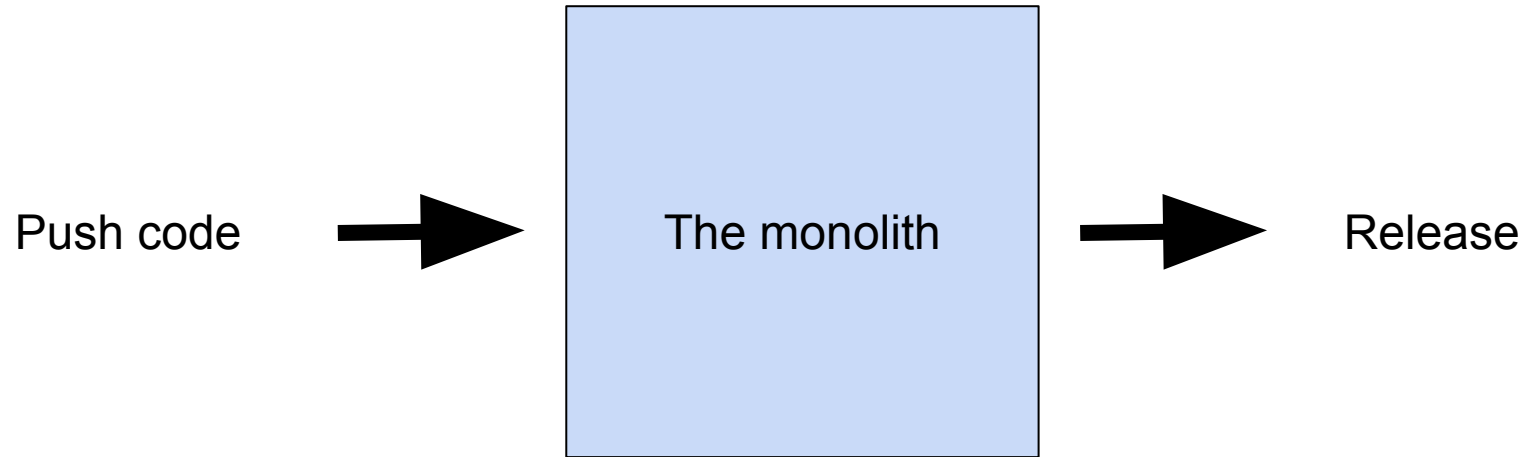
Continuous delivery

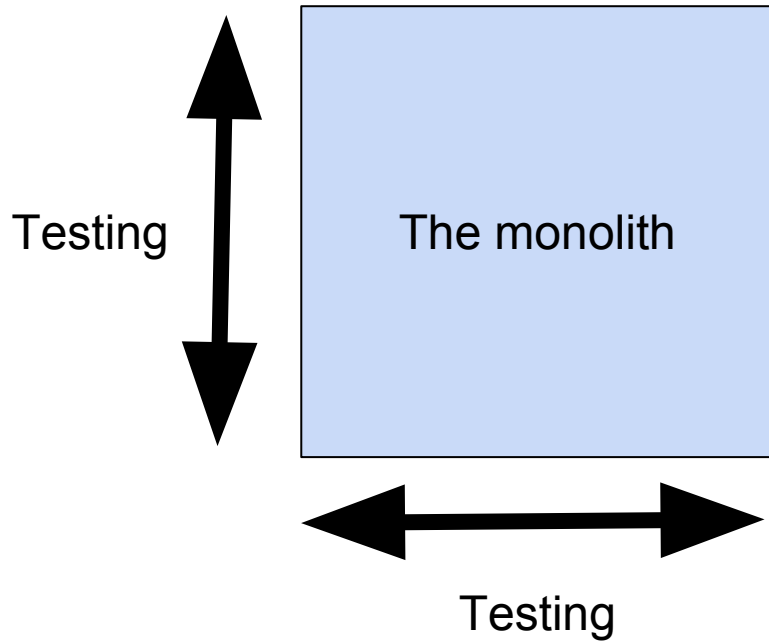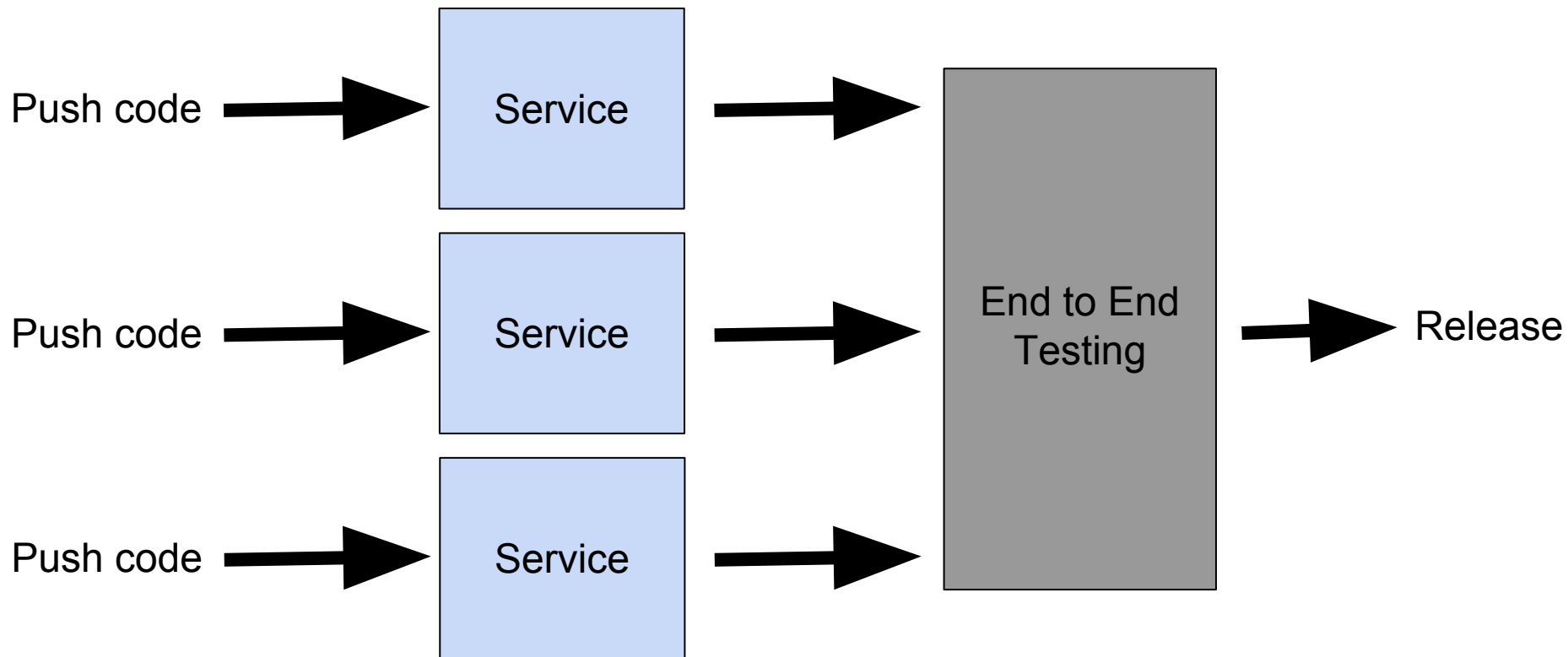# Microservices and the impact on testing

Push code → The monolith → Release
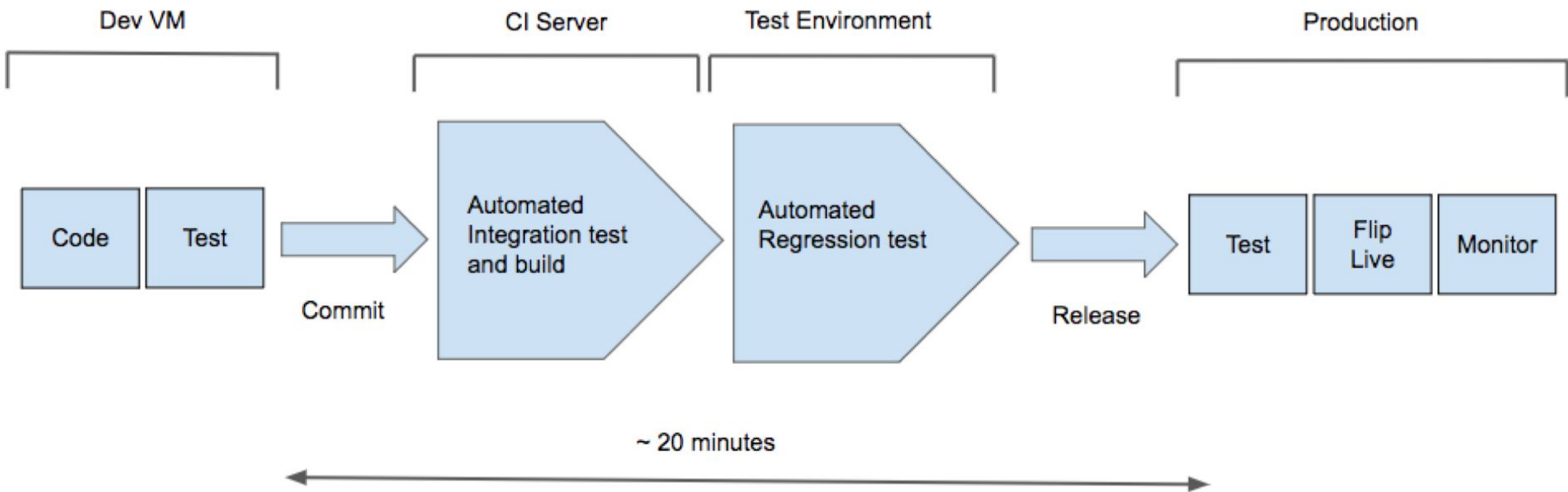
Microservices and the impact on testing

Microservices and the impact on testing

Push code → Service → Release

Push code → Service → Release

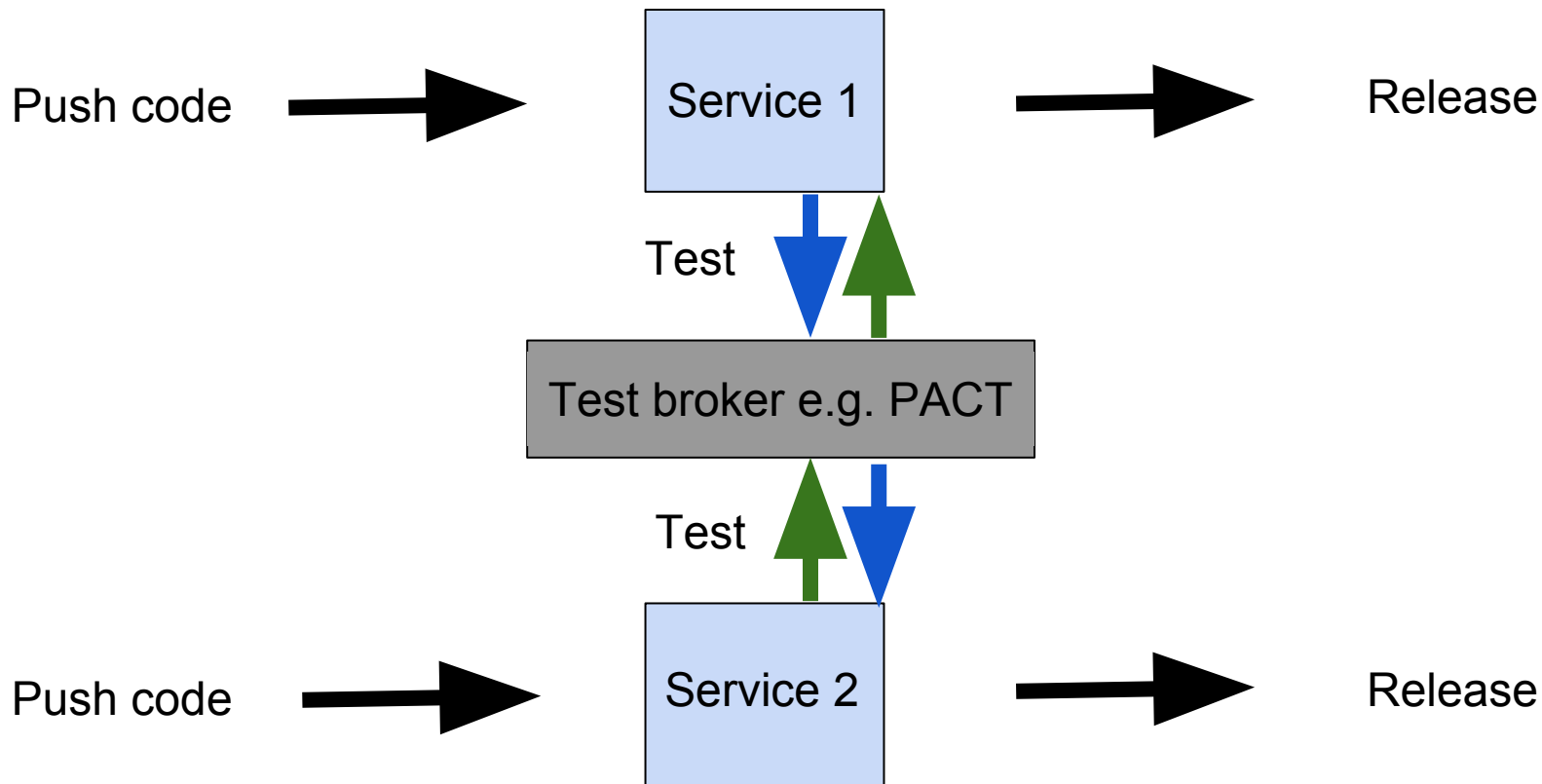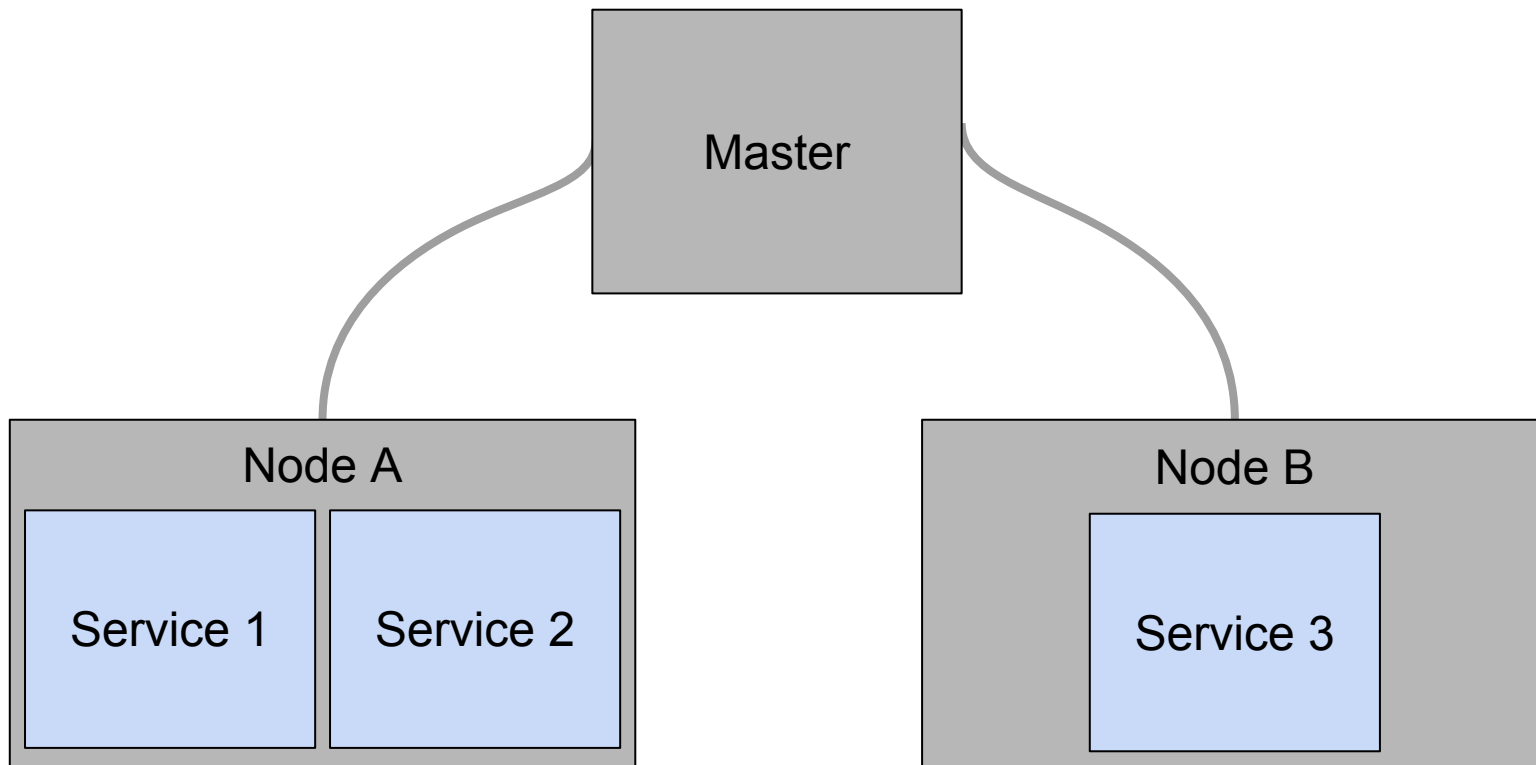Push code → Service → Release
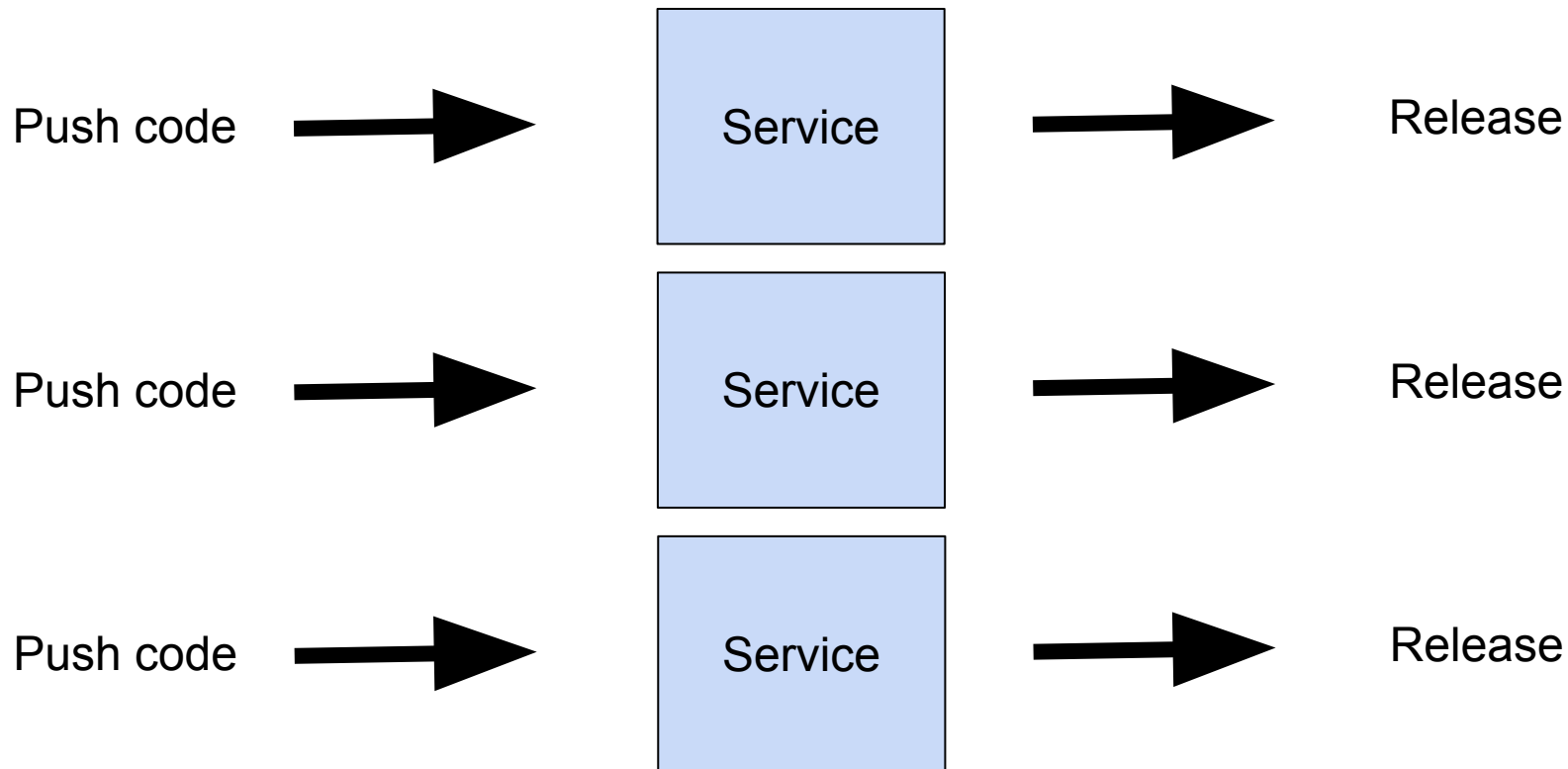
Microservices and the impact on testing
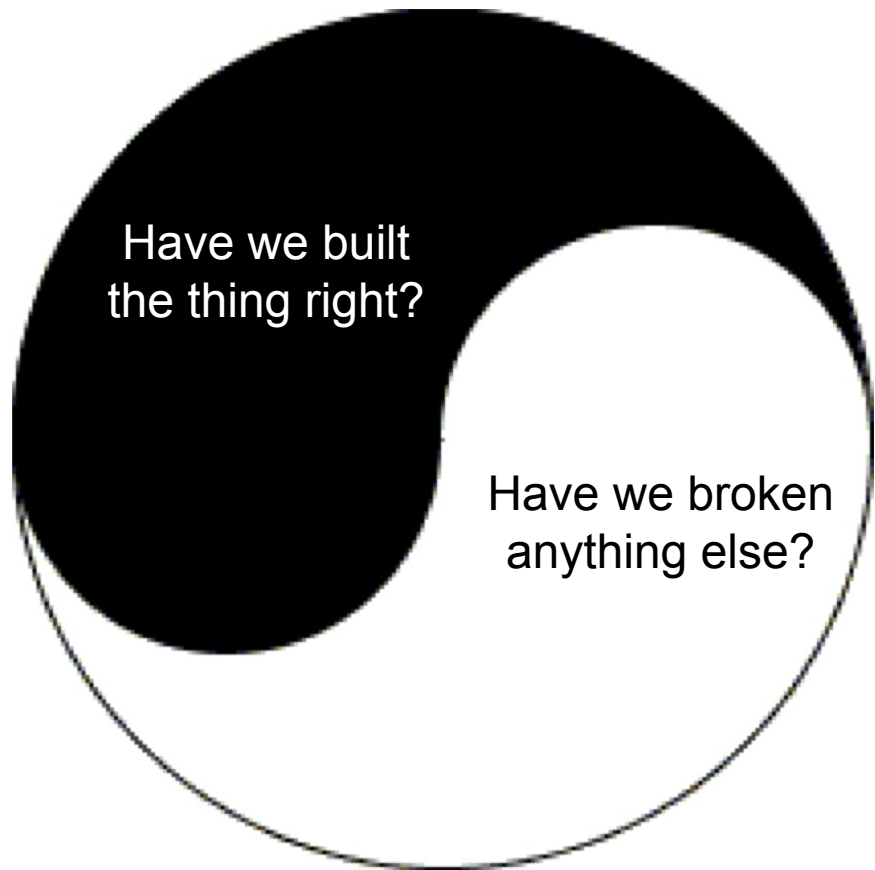
Consumer driven contract testing

Self healing systems

A quick look at testing
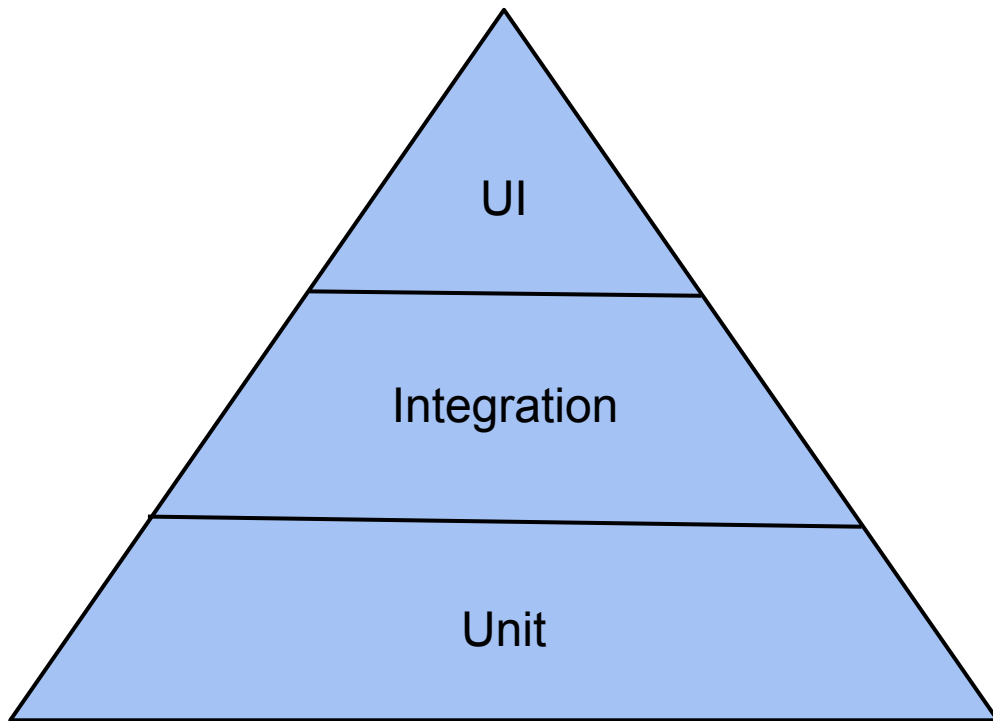
The two sides to testing

Testing

An **investigation** conducted to gather **information** on the quality of the system or product.
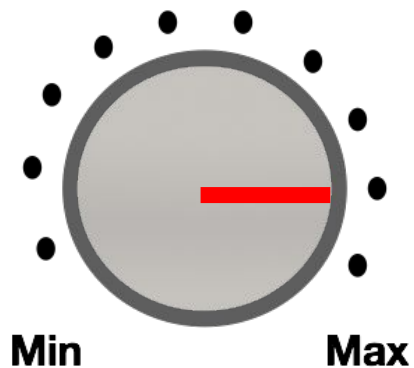
Quality
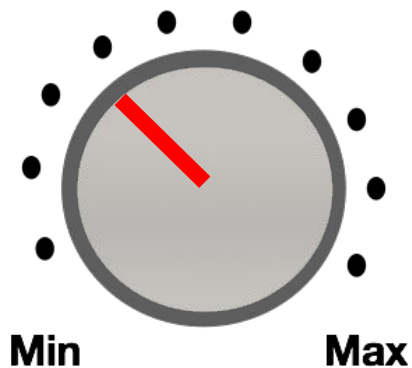
A quality system meets the **needs of the user.**
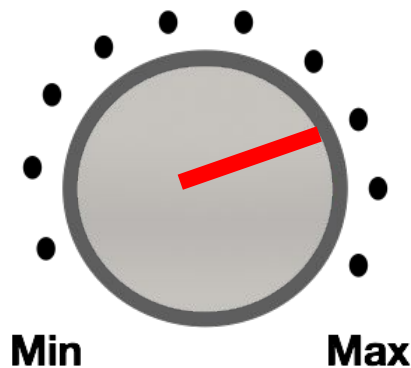
Test pyramid

Testing dials

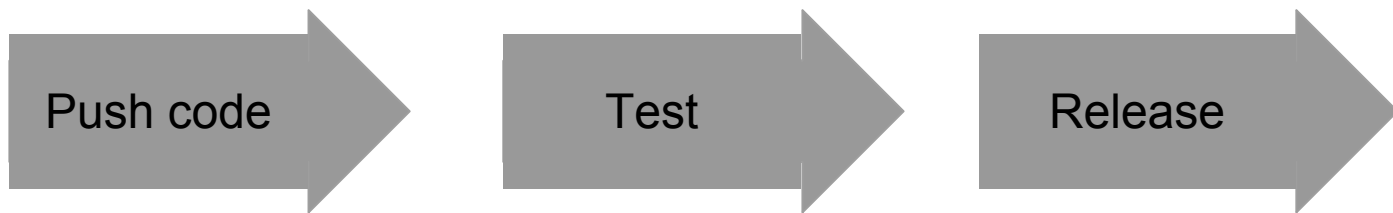Unit testing

Integration testing

Contract testing
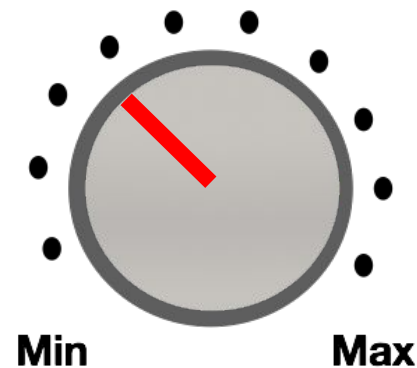
Manual testing

Testing that something works can be hard
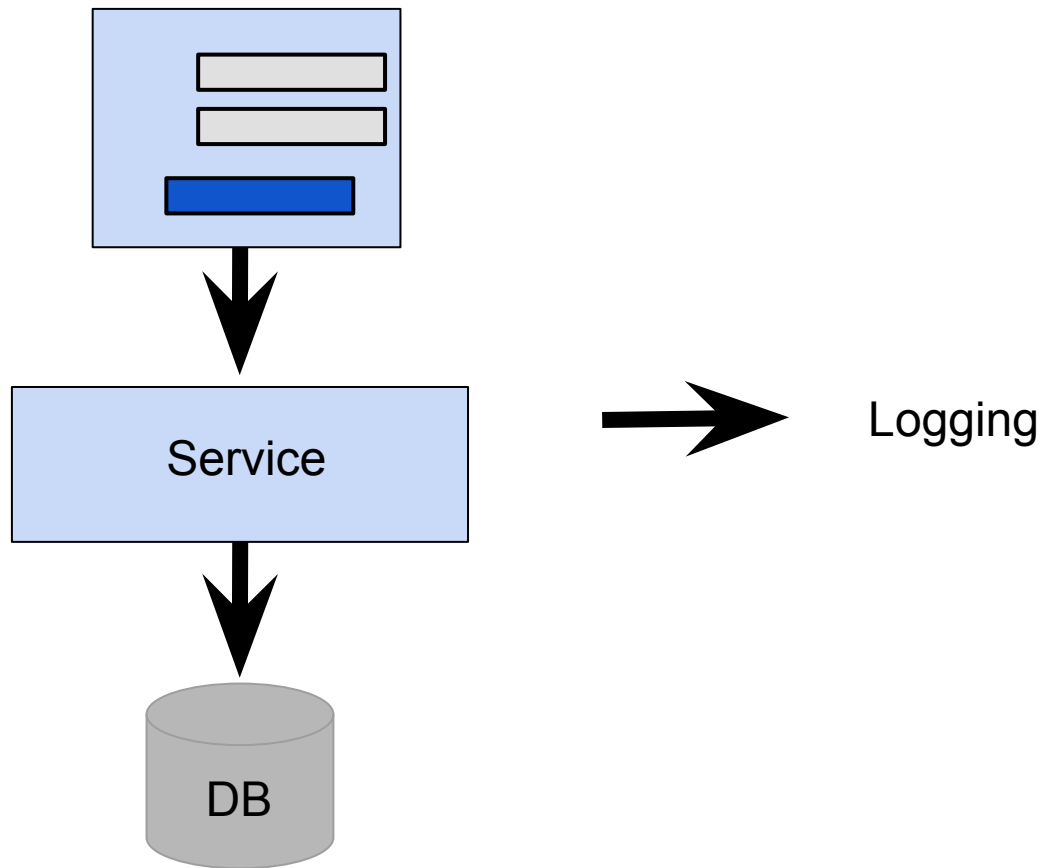
Push code

Test

Release

Two problems:
- Process assumes a positive outcome = bug reports or rework
- Forced failure states can be unrealistic = limited value

Observability instead of testing?

Retrospectively adding logging

Service

Logging

DB

How will we know when it isn't working?

How would you test observability?

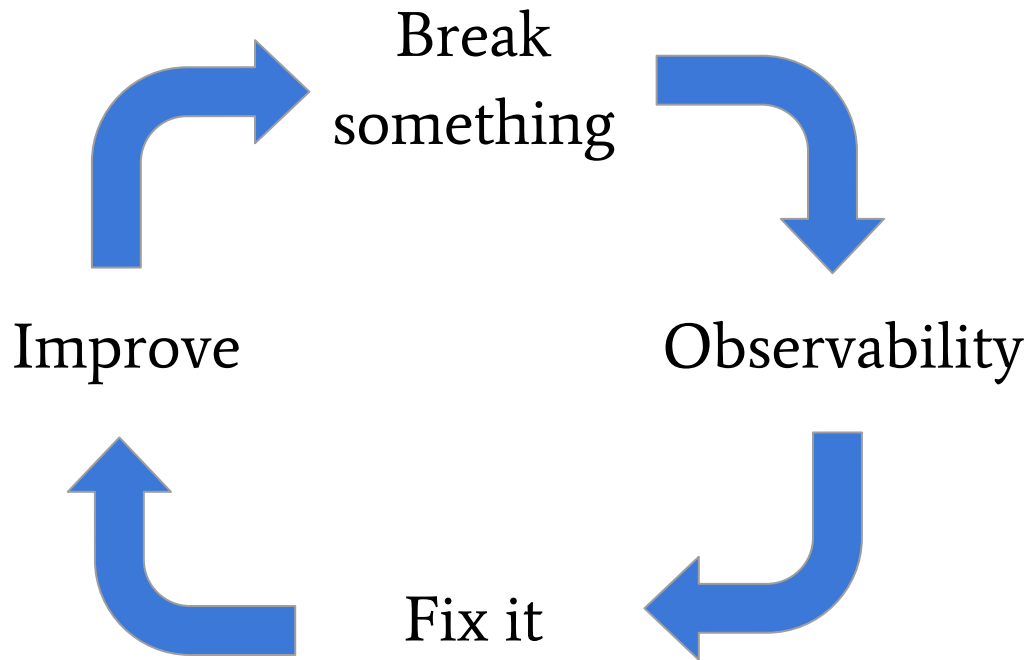|	
Monitoring	Alerting
Tracing	Log aggregation

Chaos monkey

- Weekly scheduled session
- Two "monkeys"
- One "victim"

Make it easy to observe failures
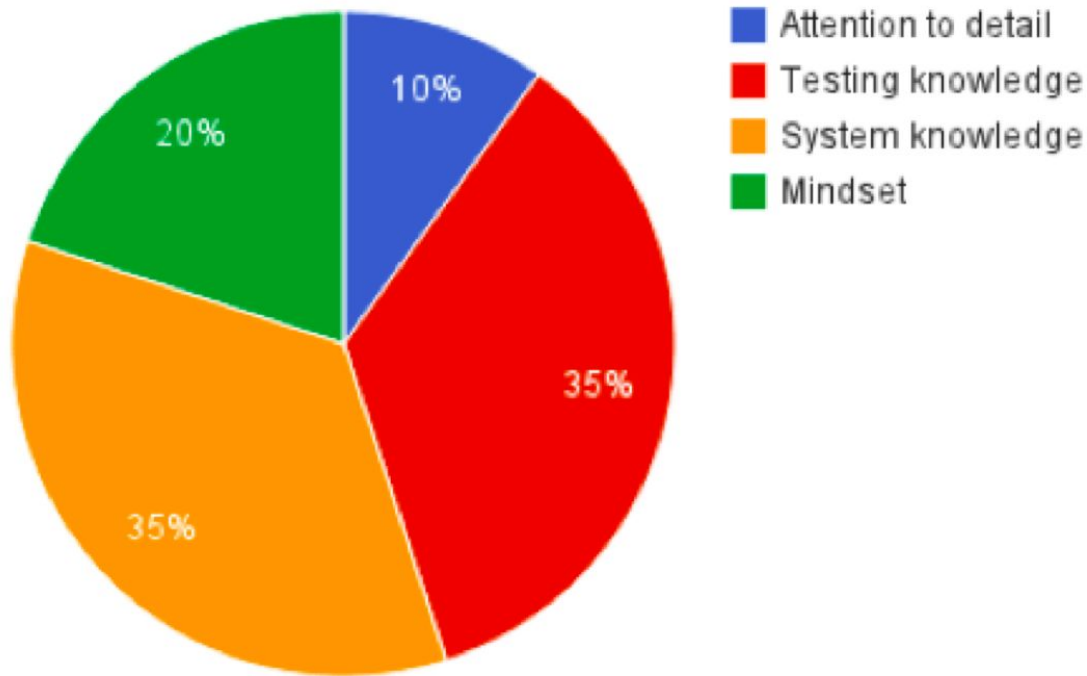
Predictable chimp

- Cheeky
- Multi-talented
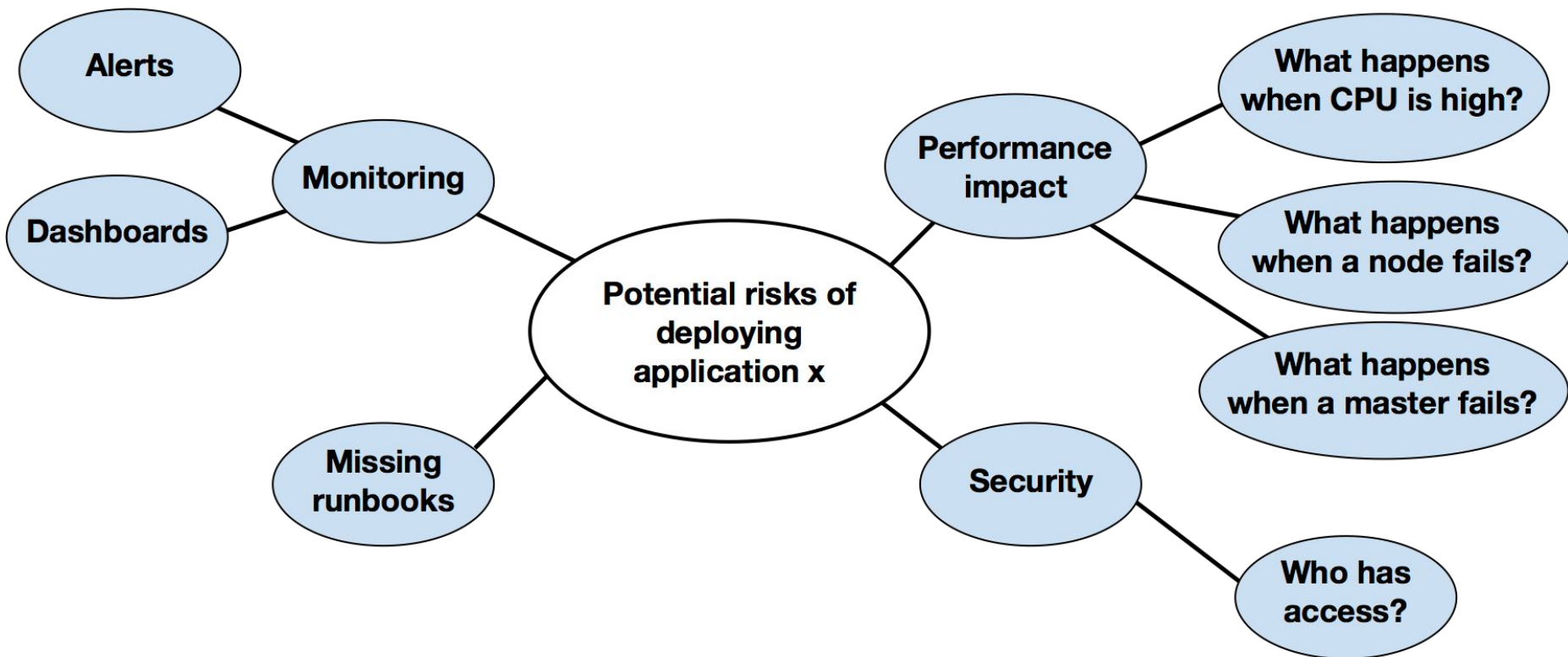- Engaging
- Attention grabbing
- Playful

http://interactivepuppy.com/spinmaster-zoomer
-chimp-interactive-monkey-toy-review/

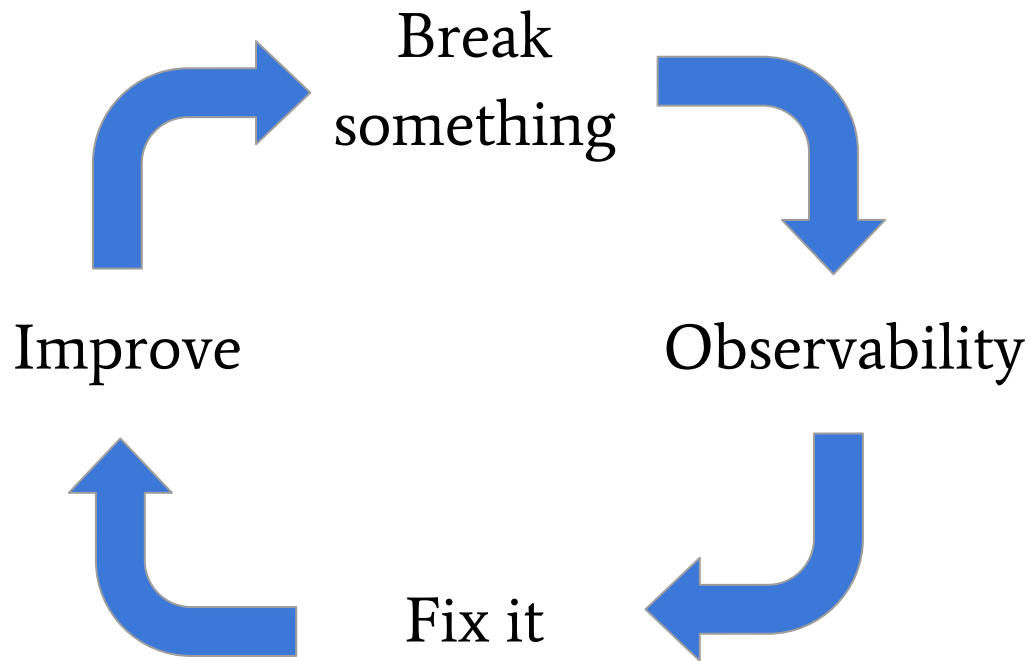# How to design test scenarios

Do you really need a tester?

Idea generation

Is it worth it?

Test value = risk vs. effort

Make it better

Break something

Observability

Fix it

Improve

In summary

- Testing is changing as systems change
- Observability may take the place of many testing tasks
- But observability isn't free
- We should be asking "How will be know when it isn't working?"

Thanks

Amy Phillips
@amyjph