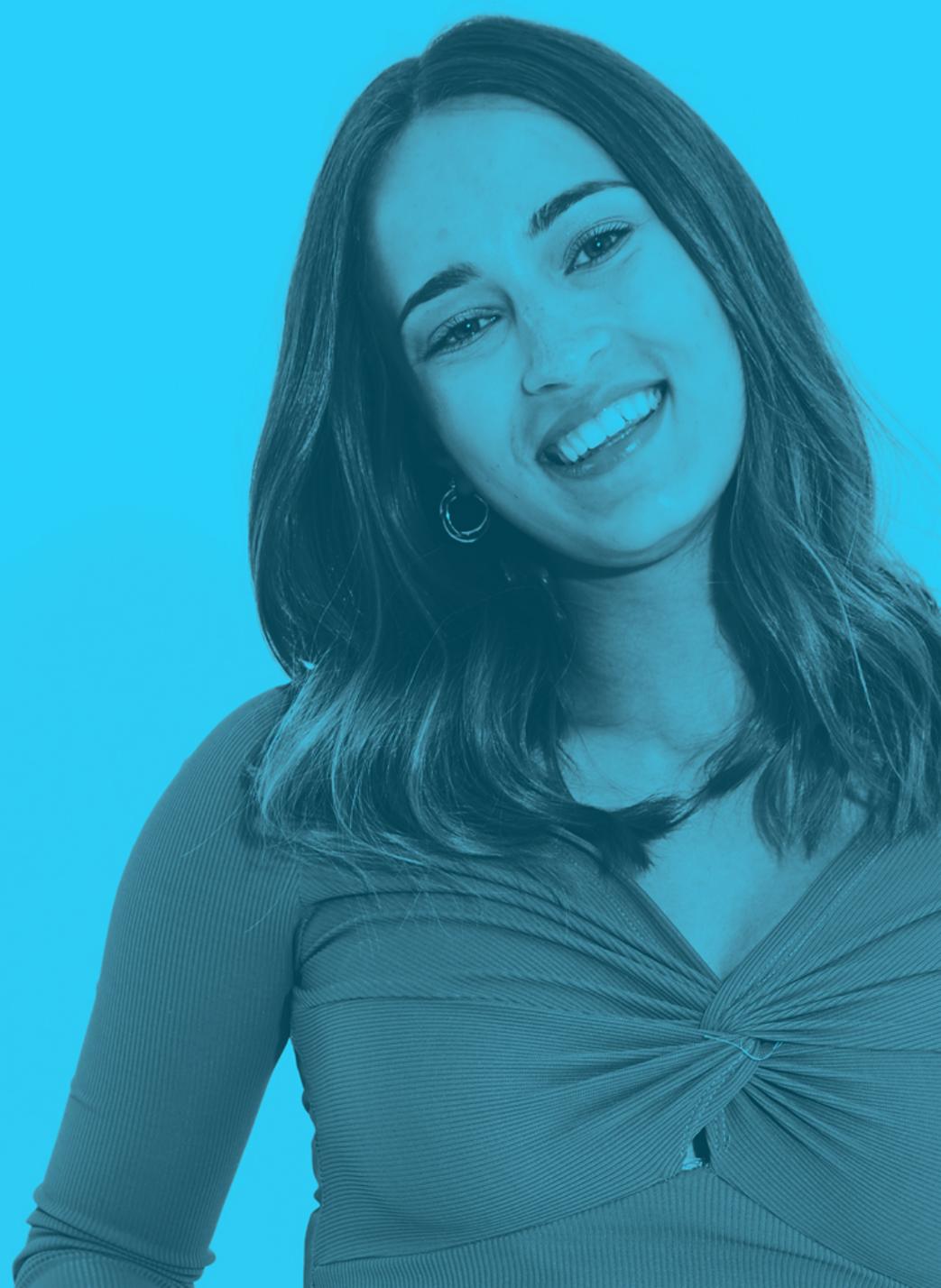
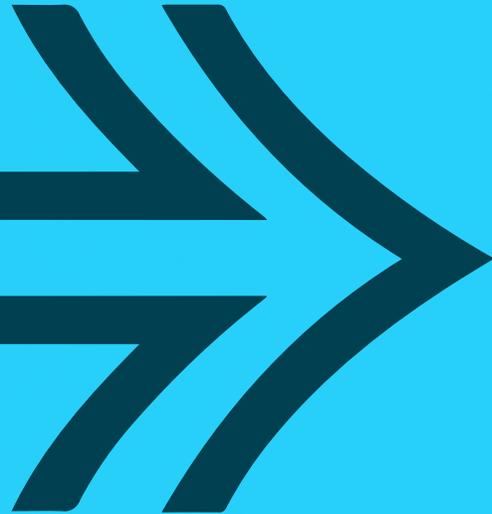




INTRODUCING DEVOPS





HOUSEKEEPING

- Fire exits
- Toilets
- Breaks



Meet & Greet

- Work in pairs
- Take 2 minutes to introduce yourselves – your name, current role, company, one of your biggest challenges at work that you want to overcome after this course and also share a Project Management tool or technique that you find useful
- Introduce your peer to the rest of the group





TOPICS

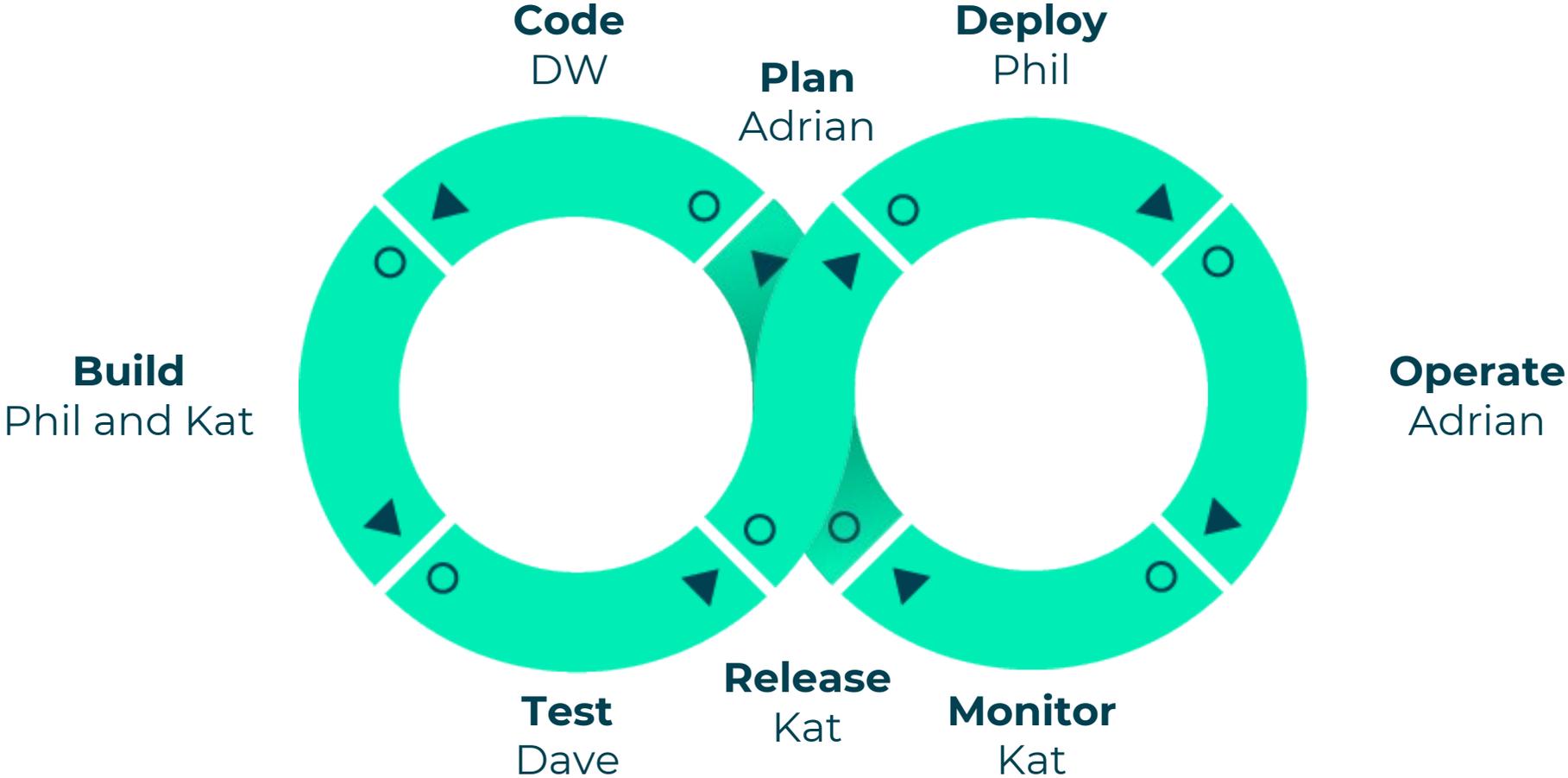
Understanding DevOps

Building a DevOps

DevOps – the tools

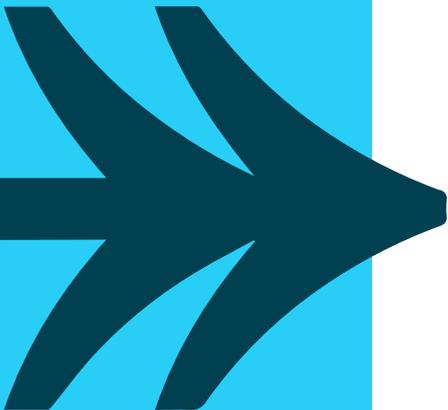


Software Development Never Ends





WHAT IS NOOPS?



NoOps stands for no operations

It is the concept that an IT environment can become so automated and abstracted from the underlying infrastructure that there is no need for a dedicated team to manage software in-house

The two main drivers behind NoOps are increasing IT automation and cloud computing

At its most extreme, a NoOps organisation is one that has no operations employees at all, however various other systems can be referred to as 'NoOps' as well



WHAT IS WEBOPS?



WebOps, short for Web Operations

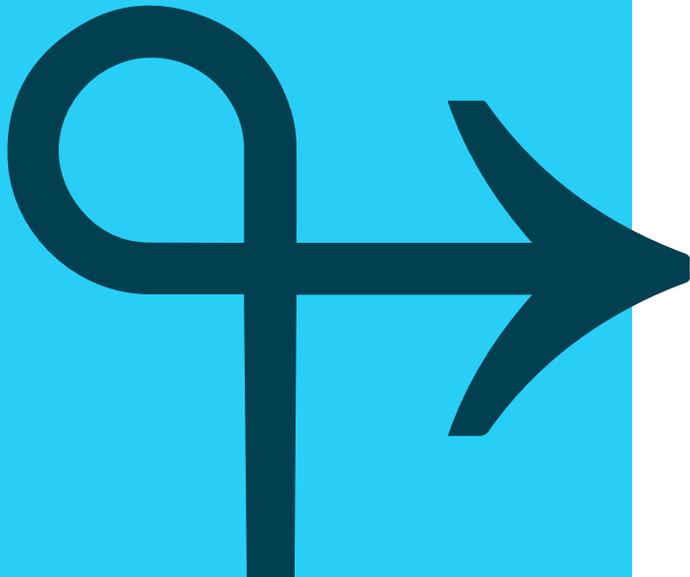
- Is the area of IT systems management that deals with the complexities of Web-based applications and the systems that support them

The area of WebOp engineering includes:

- Application deployment
- Management
- Maintenance
- Configuration
- Repair



DEVOPS IS A NEW KIND OF ROLE

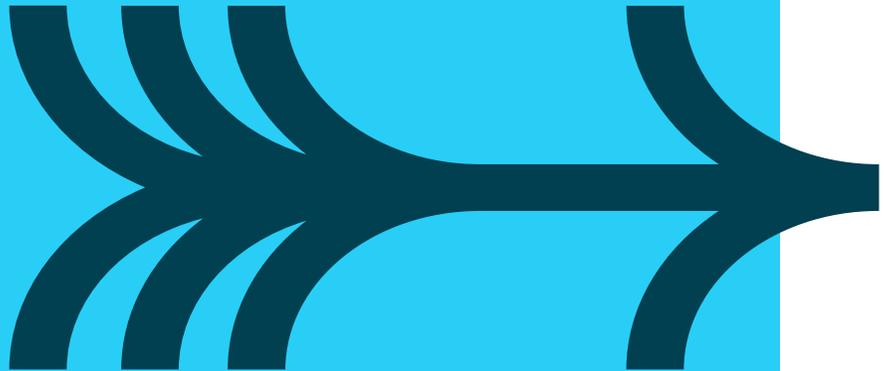


Extend existing skills to work in a more Agile way





FROM THE DEV'S PERSPECTIVE



Developers that can maintain infrastructure

Continual delivery

Continuous integration

First Deployment

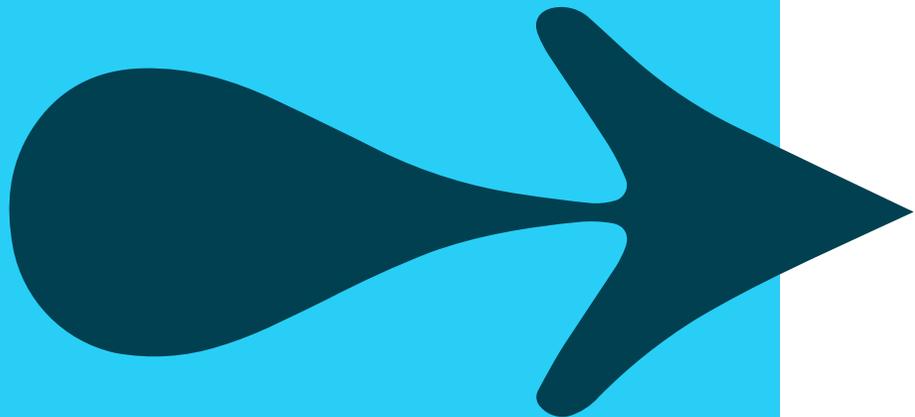
Monitored

Tested





FROM THE OPS PERSPECTIVE



Ops staff that can code

Virtualisation

Cloud based

High availability

Testable

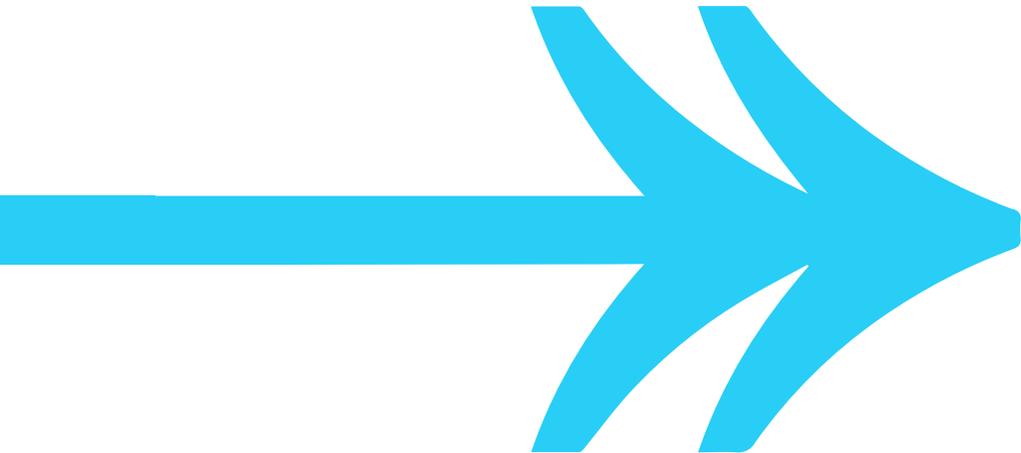
Maintainable

Reproducible





From the QA / Tester Perspective



Testing as a shared responsibility

Testable

Monitored

Reproducible

Version managed

Maintainable





From the BA / PM Perspective

Data that matters; visibility, improvement

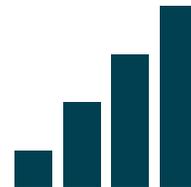
High availability

Maintainable

Reproducible

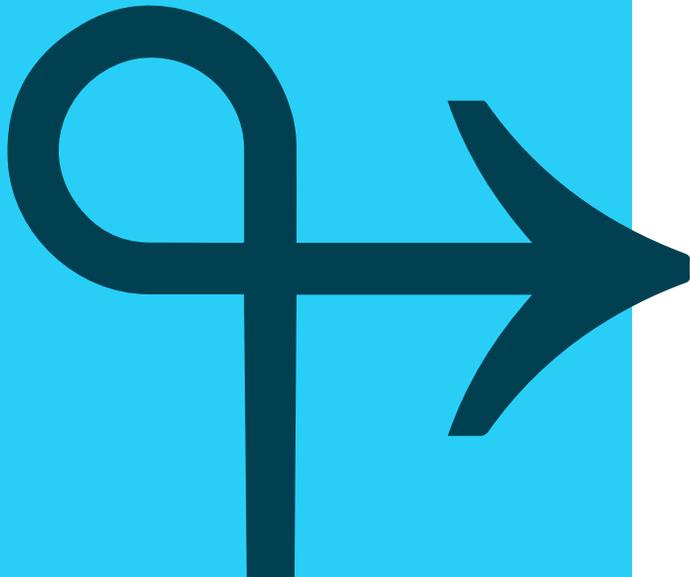
Continuous improvement

Measurable

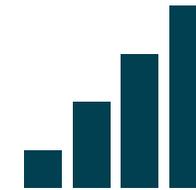
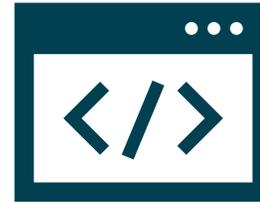




DEVOPS TEAMS WORK TOGETHER

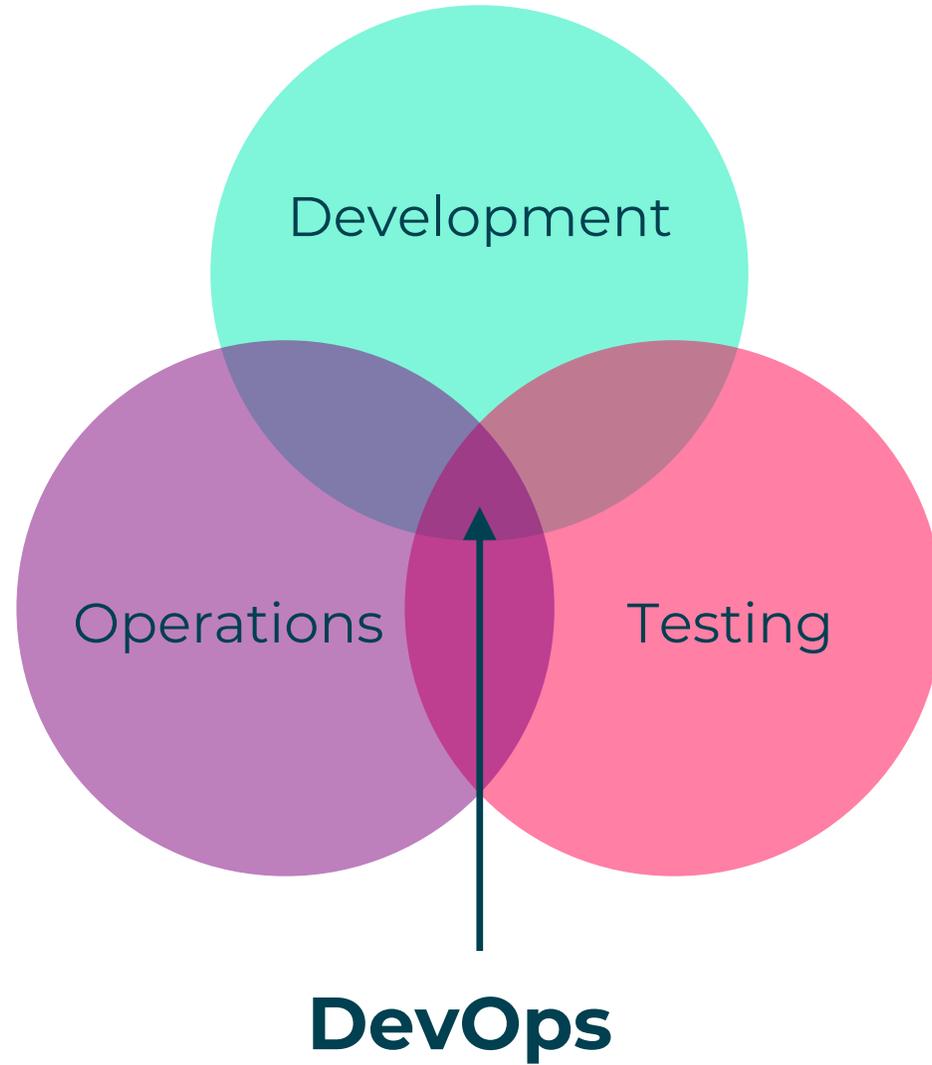
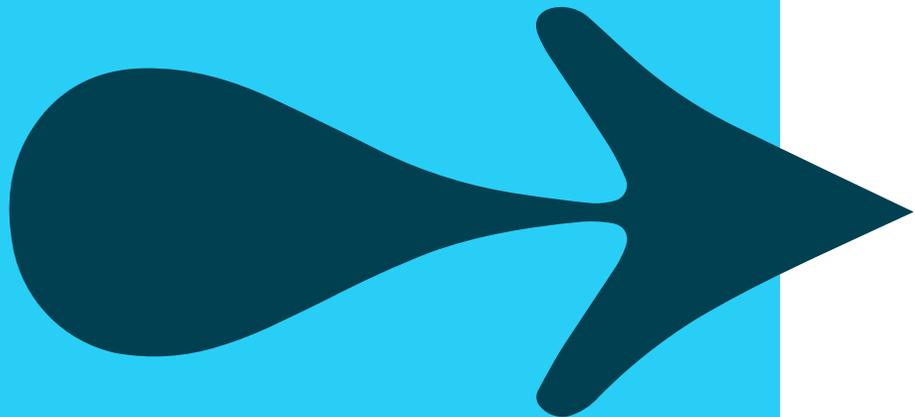


One Team – One Goal



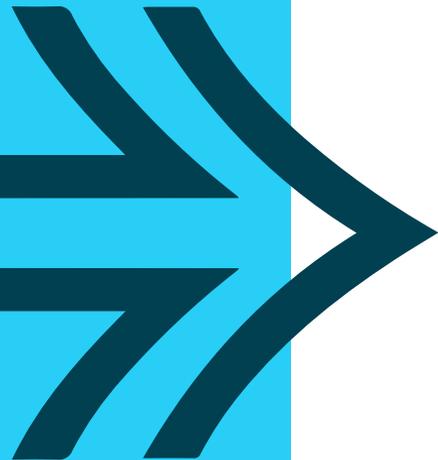
QA

**DEVOPS -
WE'RE ALL IN
THIS
TOGETHER!**





KEEP CALMS AND DO DEVOPS



DevOps – The C.A.L.M.S Lifecycle:

C – Culture

A – Automation

L – Lean

M – Measurement

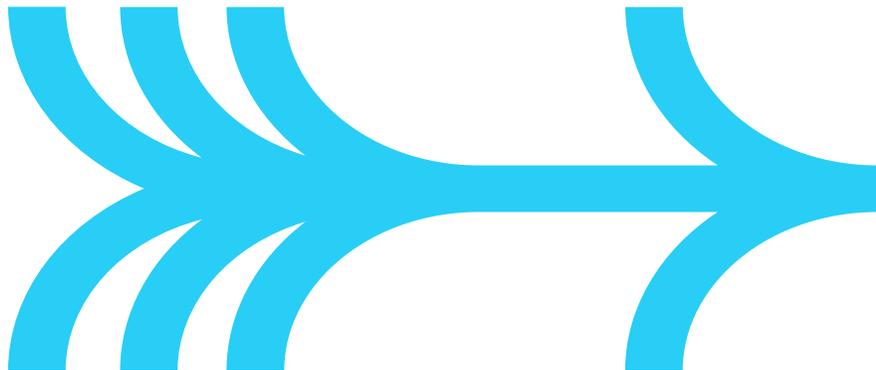
S – Sharing



C – DevOps Occurs Through Culture

“It’s amazing what you can accomplish if you do not care who gets the credit.”

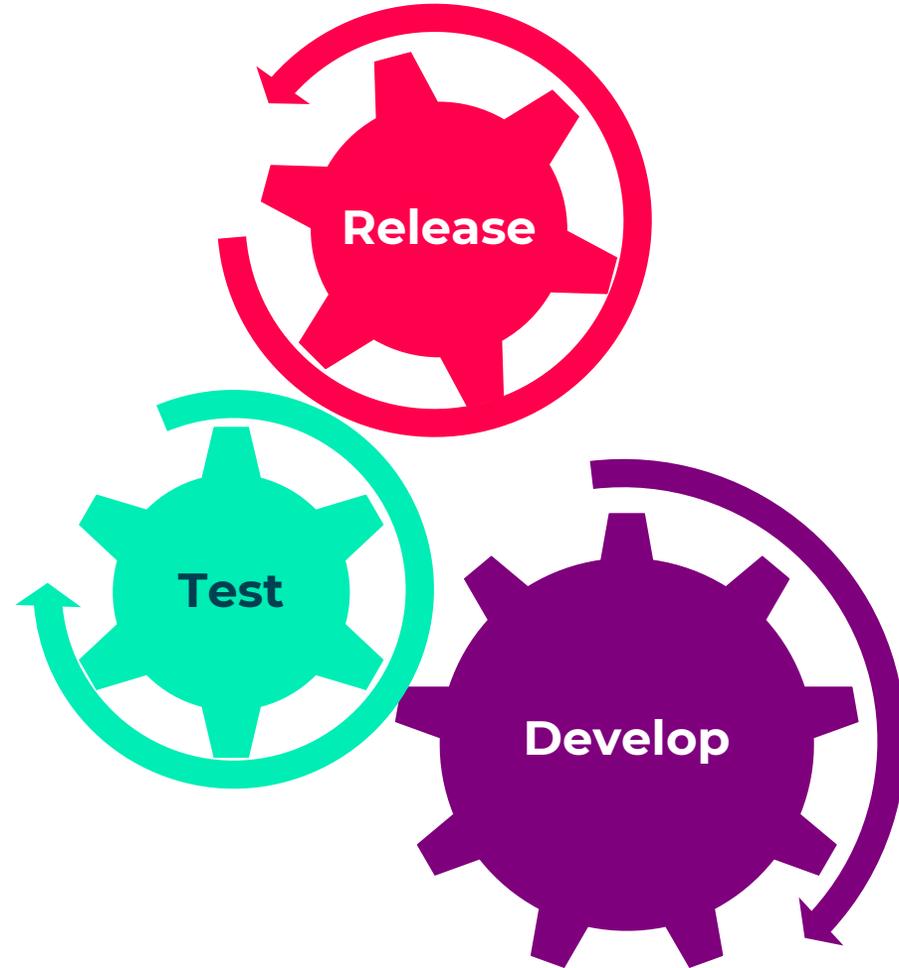
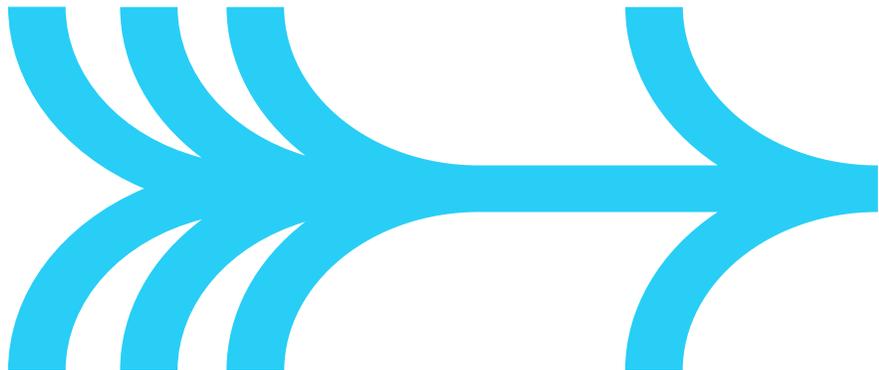
President Harry S. Truman





A – Automation

Tools make automation possible





L - LEAN - WORKS WITH AGILE



1. Eliminate Waste
2. Build Quality In
3. Create Knowledge
4. Defer Commitment
5. Deliver Fast
6. Respect People
7. Optimise The Whole

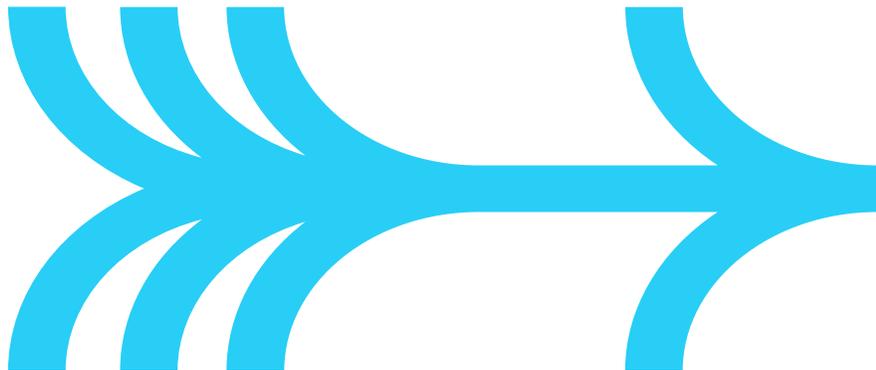
Mary and Tom Poppendieck



M – Monitoring

“Don’t it always seem to go, that you don’t know what you’ve got ’til it’s gone.”

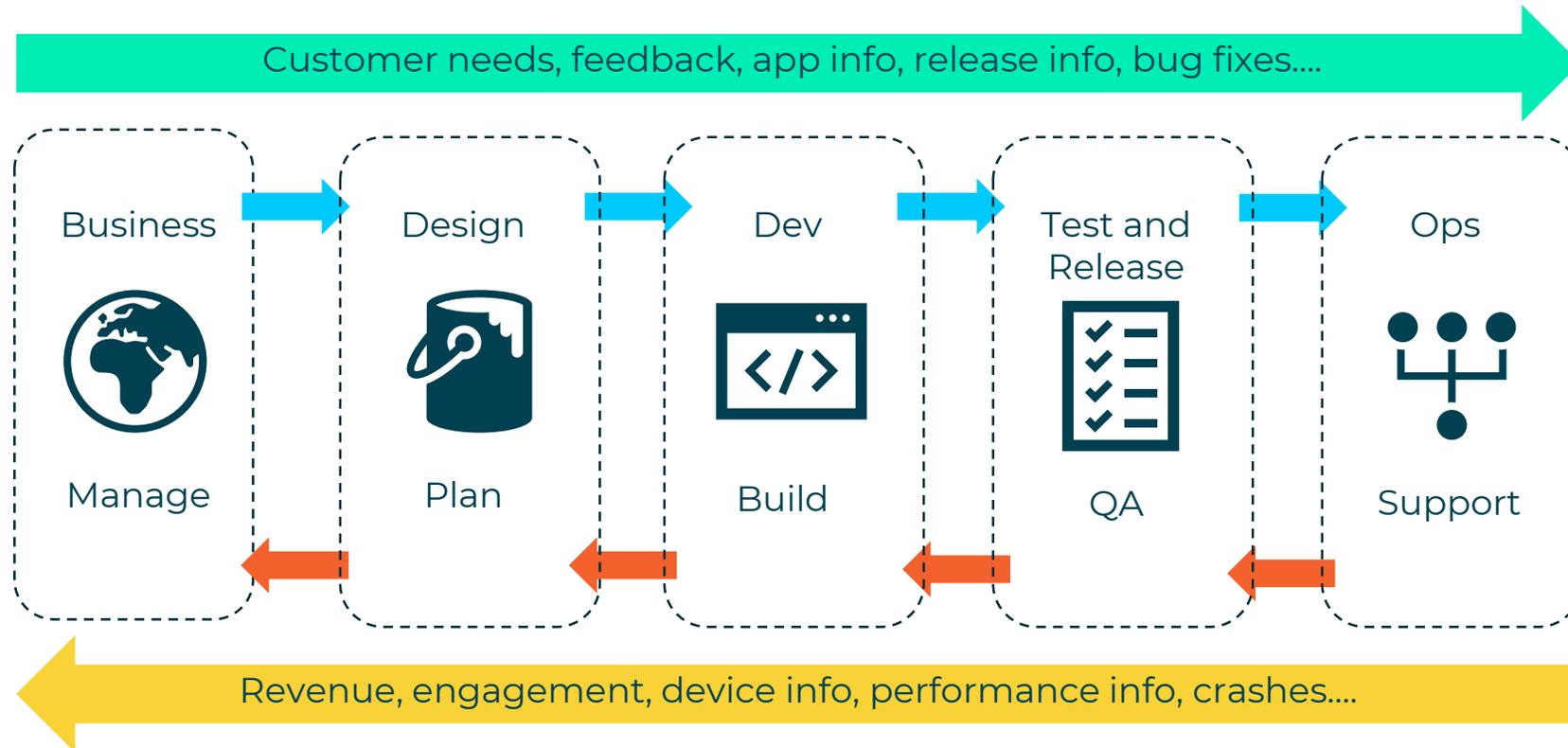
Joni Mitchell, *Big Yellow Taxi*



QA S - Sharing

“Knowledge is experience, everything else is information.”

Albert Einstein





Demonstration – Lean Coffee

Created by Jim Benson and Jeremy Lightsmith in 2009

Lean Coffee allows us to host meetings without agenda:

- Structured through democratic engagement
- People talk about what they want to talk about

It follows a simple flow:

- Set up a personal Kanban
- We note what we want to discuss
- We vote and talk
- Each activity is time-boxed



Better
Faster
Stronger



BUILDING A DEVOPS CULTURE

Culture

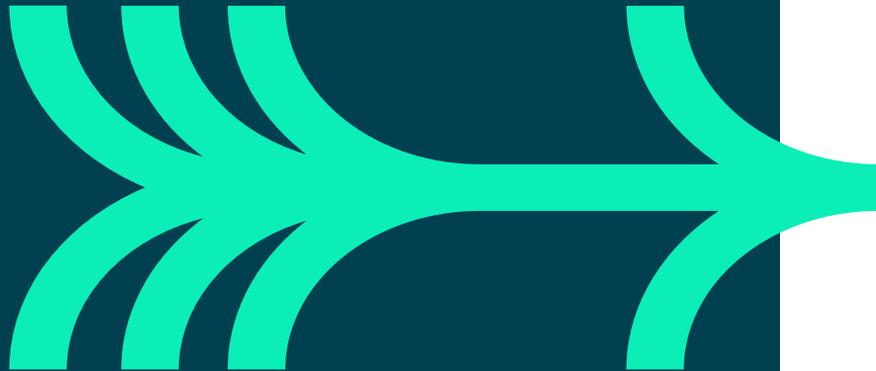




**Amazon push code
to production
every 19 seconds**



WE NEED ROUNDED WEB PROFESSIONALS



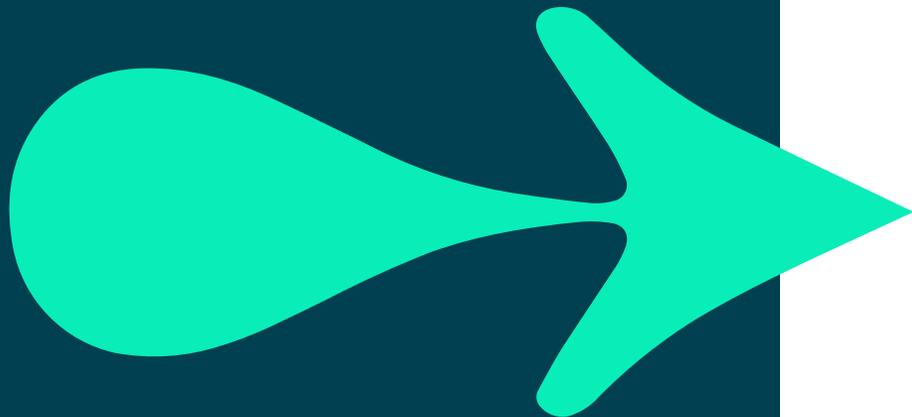
The agile world demands multifaceted teams

- No-one working in development can choose not to be involved end-to-end
- This is true of the wider





WHAT STOPS US?



“It’s too complicated.”

“I’ll break it.”

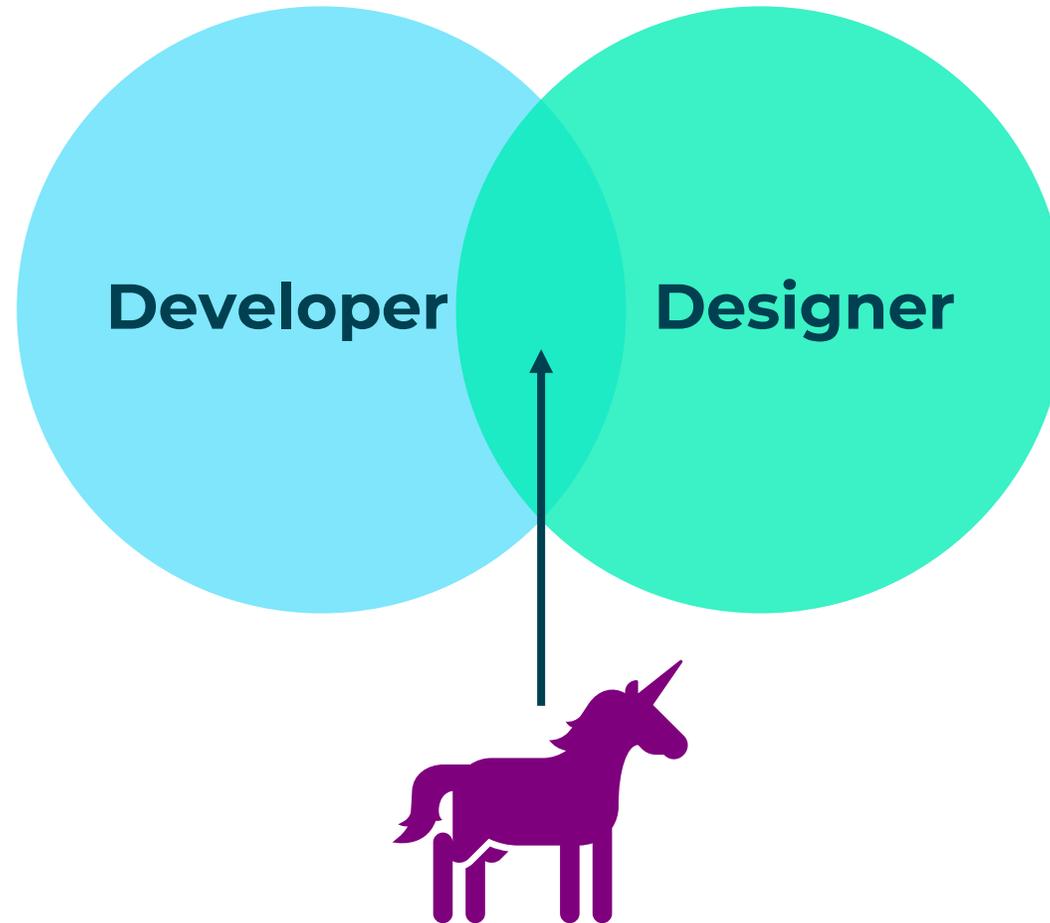
“What’s with all this code?”

“What is GitHub?”

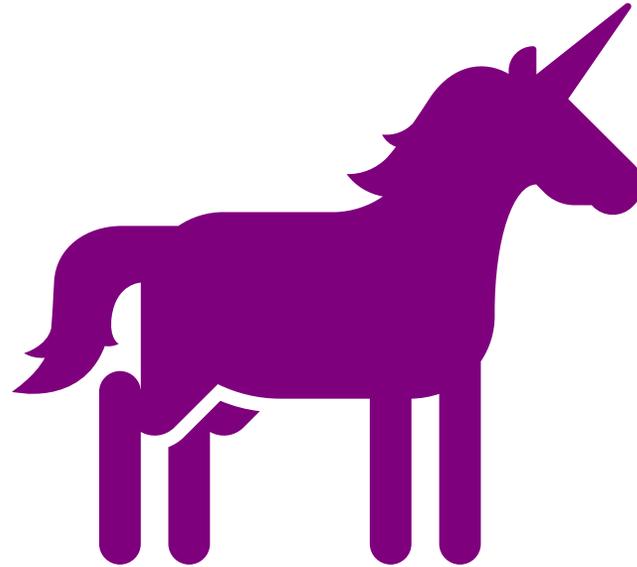
“That’s not my responsibility.”

Reasons why organisations fail to change

QA The type of Skills Required are Different



QA The type of Skills Required are Different



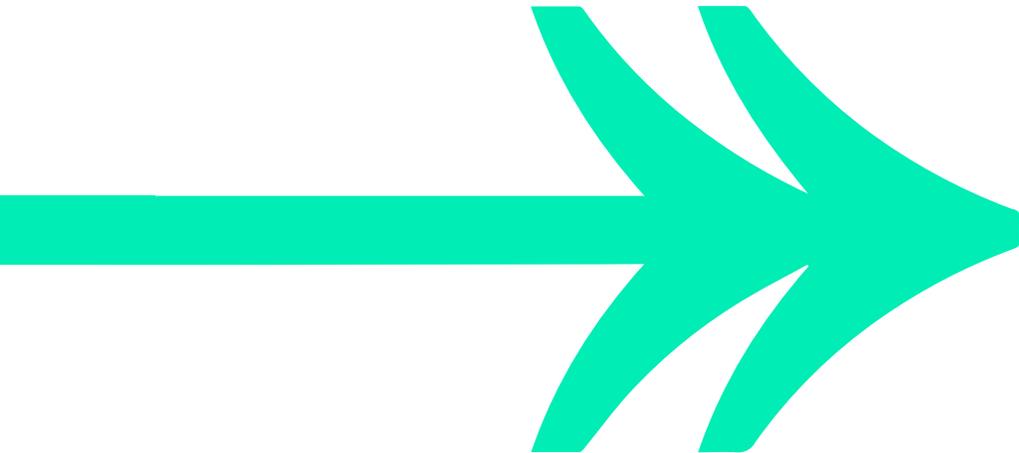
BE THE UNICORN



DevOps is Collective Responsibility

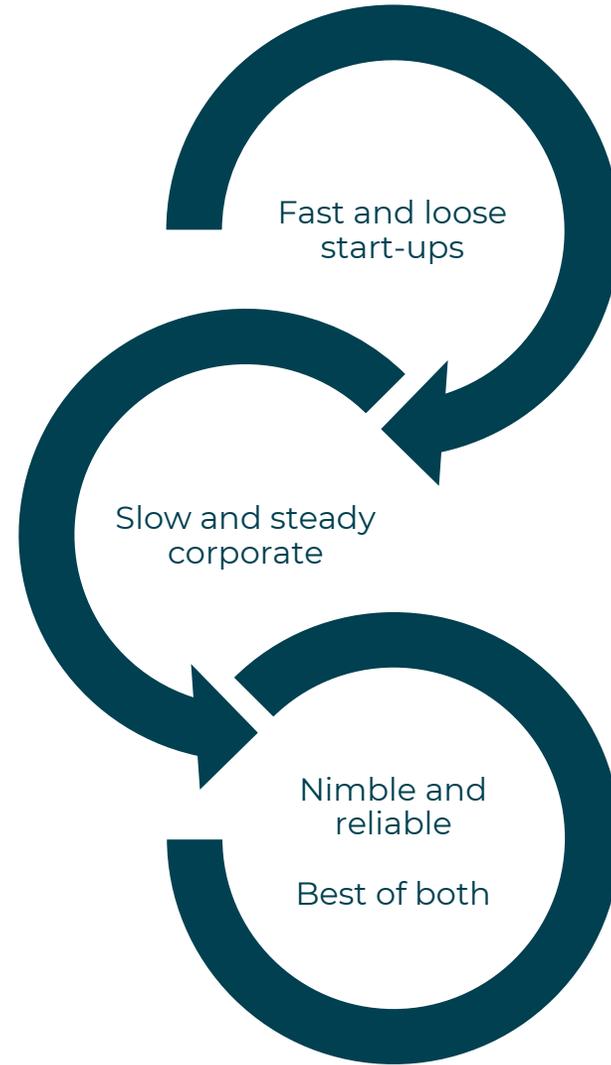
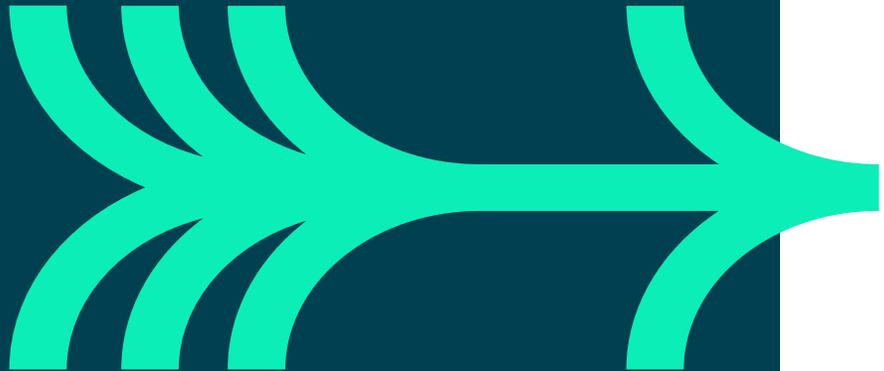
“It's not the tools that you have faith in – tools are just tools.
They work, or they don't work. It's people you have faith
in or not.”

Steve Jobs



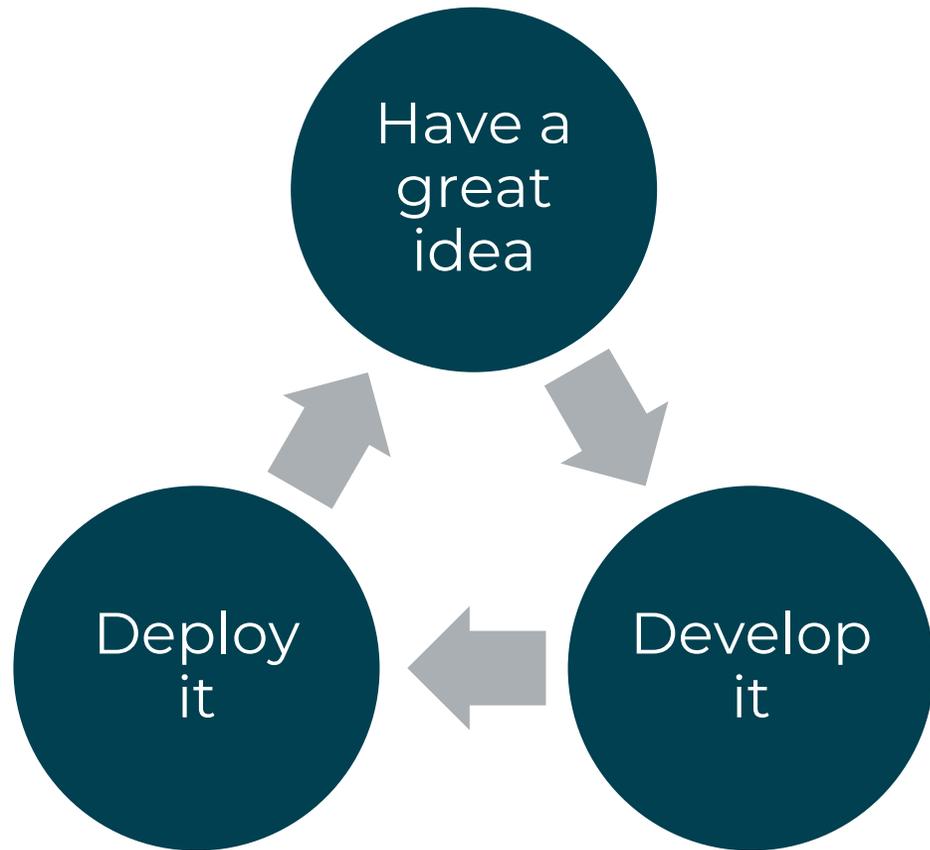


WHAT PROBLEM DOES DEVOPS TRY TO SOLVE?

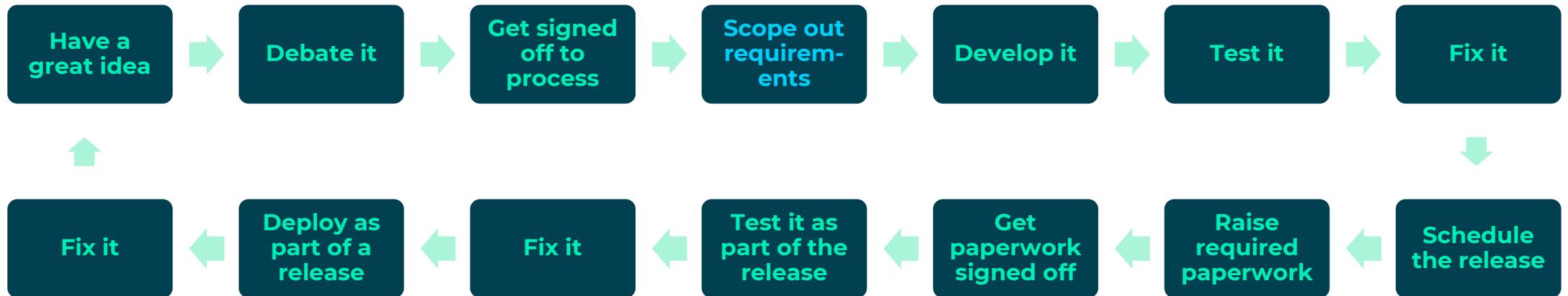




START-UPS – GREEN FIELDS AHOY!



QA Enterprise – Complexity Due to Scale





Continuous Integration

QA Seven Wastes of Lean*

Working in pairs, identify examples of each of the wastes from your own experience.

Partially-done work	
Extra processes	
Extra features	
Task switching	
Delays	
Handoffs	
Defects	

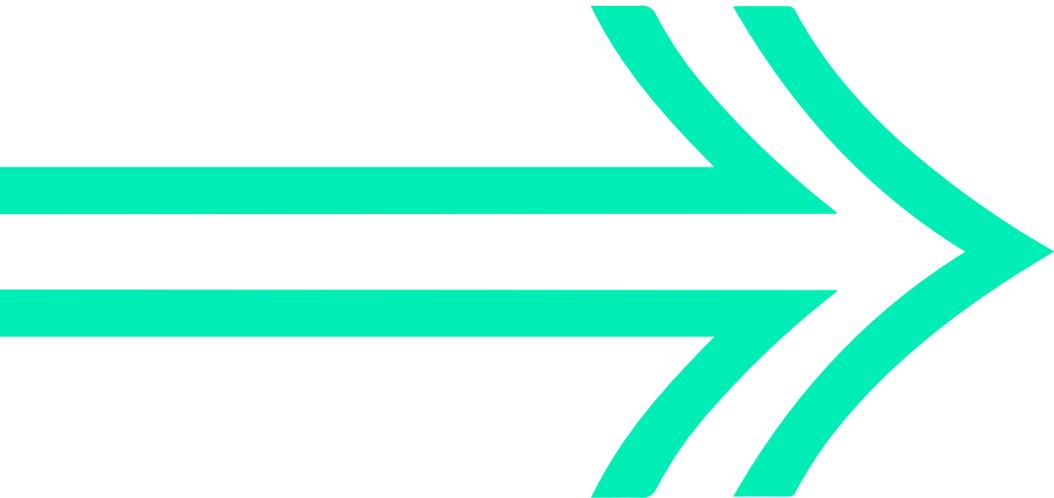
* Adapted for software development from traditional lean



LEAN and Kanban

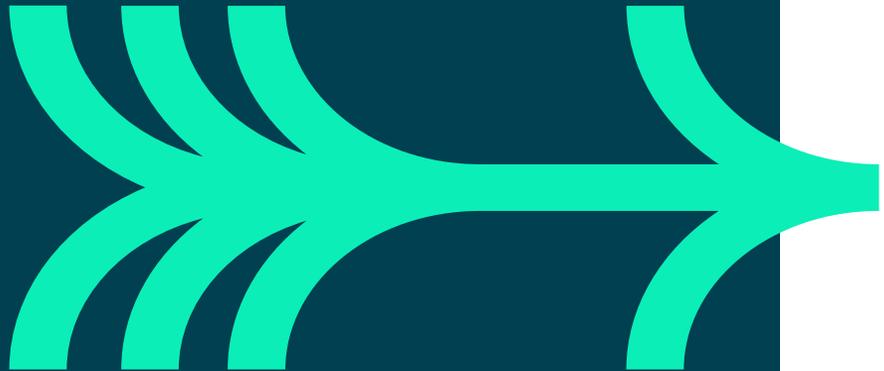
“The aim of Kanban is to make troubles come to the surface and to link them to a Kaizen Activity.”

Talichi Ohno





USE AGILE AND LEAN TO DEMONSTRATE PROGRESS



Regular communication

New information, changing priorities, progress updates are communicated daily between the business owner and the implementation team

The team meets every day in the same time and place
– the daily stand-up meeting

Improve Collaboratively

“If egoless programming is used, everyone in the group will have the opportunity to examine the work of everyone else at the same time.”

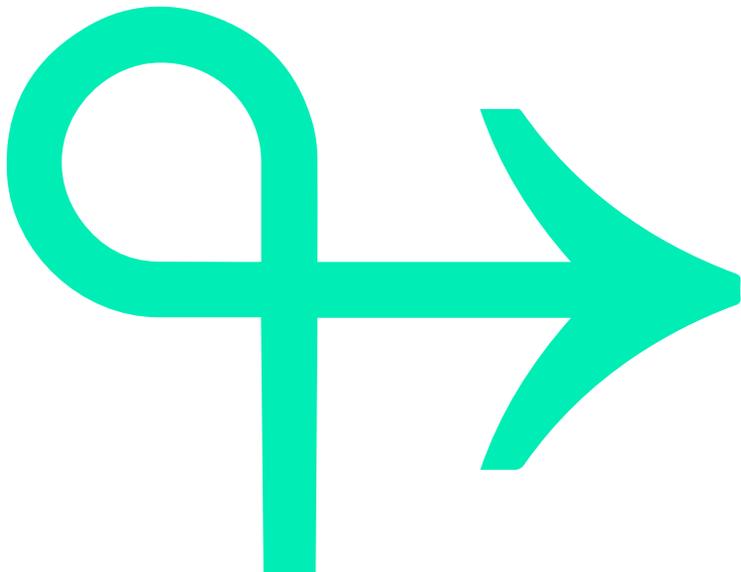
Gerald Weinberg



Improve Collaboratively

“If egoless programming is used, everyone in the group will have the opportunity to examine the work of everyone else at the same time.”

Gerald Weinberg





Encourage Openness



RECAP



Behavioural Checklist

Visualise What You Do

Limit WiP

Manage the Flow



CONTINUOUS IMPROVEMENT



Continuous improvement

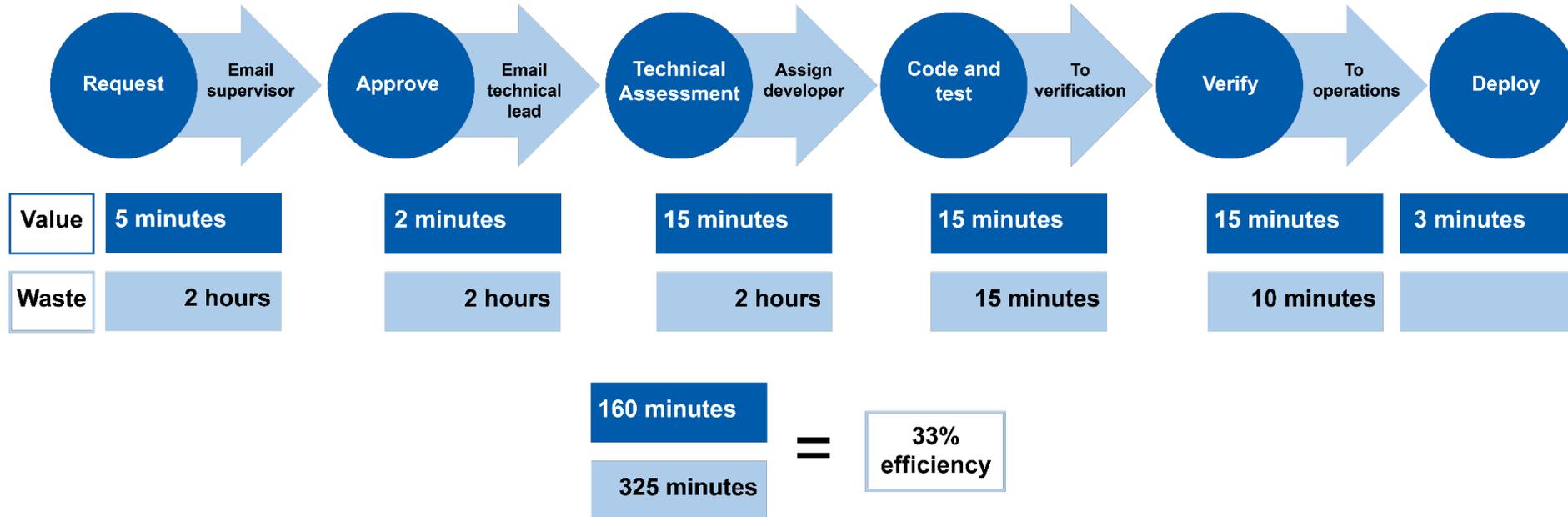
- Kaizen – 'good change' – continuous improvement mindset

Targeting areas for improvement

- Bottlenecks
- Waste elimination
- Reduction of variability



Example – Value Stream Map

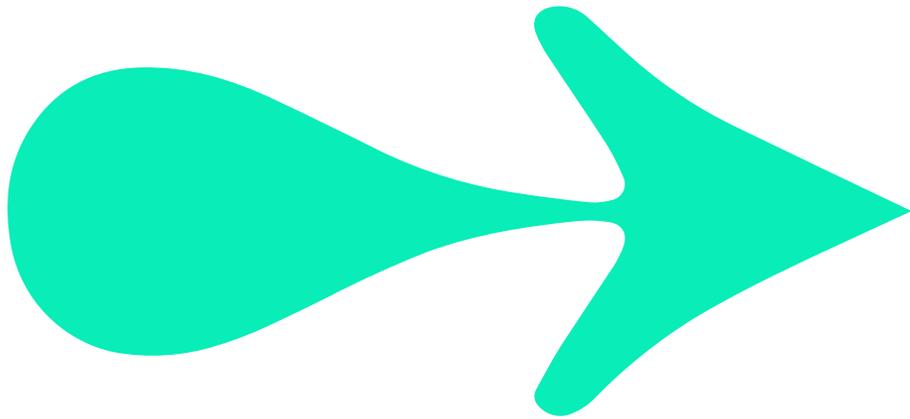




Evolve Experimentally

“It is important that instead of processes being adopted, it is adapted.”

Jutta Eckstein

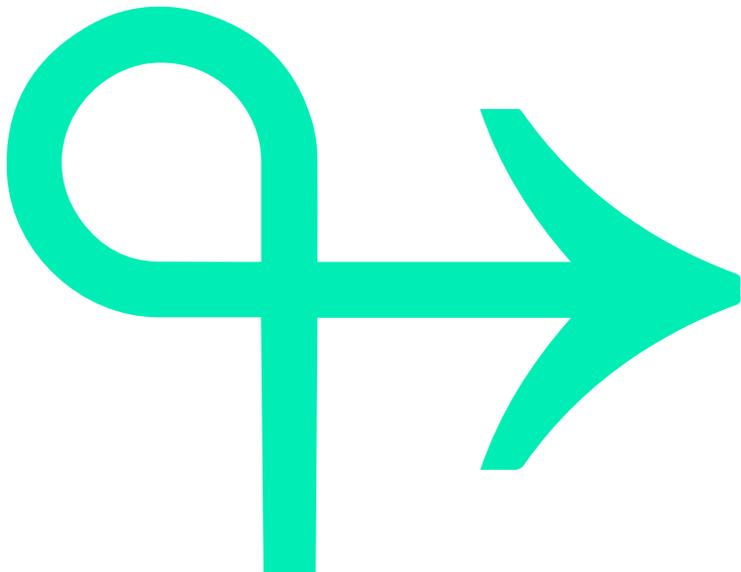




Implement Feedback Loops

“Ignoring feedback merely means that the system will eventually experience a massive unpleasant surprise rather than a small unpleasant surprise.”

John Gall

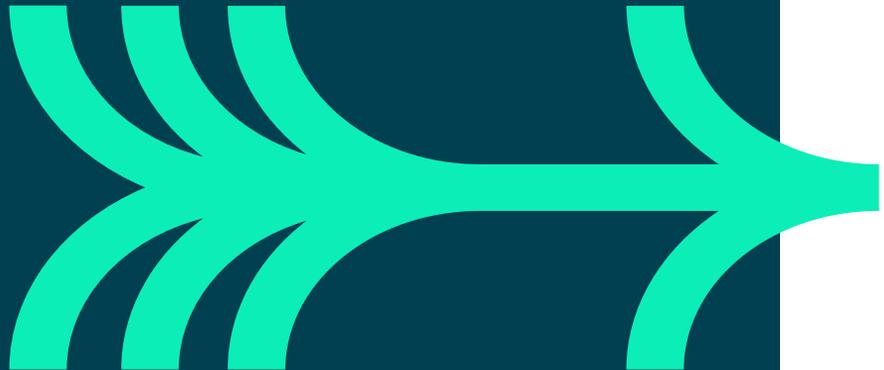




Netflix Simian Army and Testing on the Live System



A LESSON FROM NETFLIX



Simian Army

- A collection of chaos monkeys that reap havoc on their production systems

A new approach to architectures – microservices

- Teams must clearly understand what service failure means to the system as a whole
- Live service testing is becoming a new norm – Netflix simian army used in the day on datacentres and services to test resilience and monitoring
- A lot of real-time monitoring
- Detect – e.g. requests per second, and transactions per second
- Respond
- Performing “semantic monitoring”
- Run subset of automated tests against live system



WE USE TOOLS FOR REASONS

Automation

Orchestration

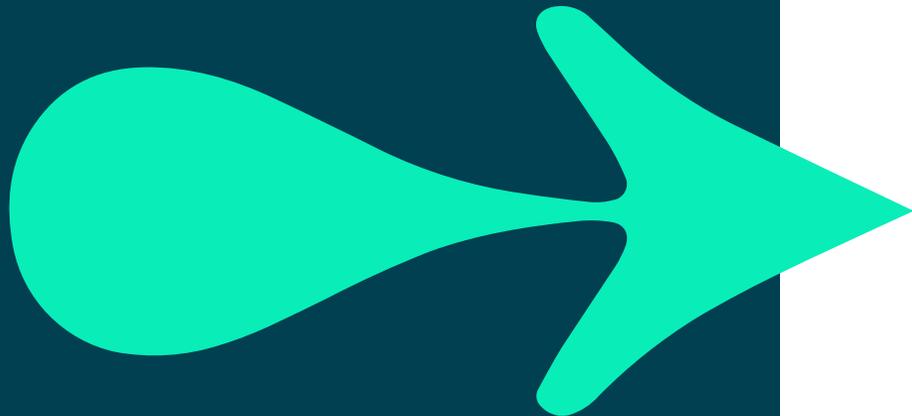
Configuration

Testing

Logging and monitoring

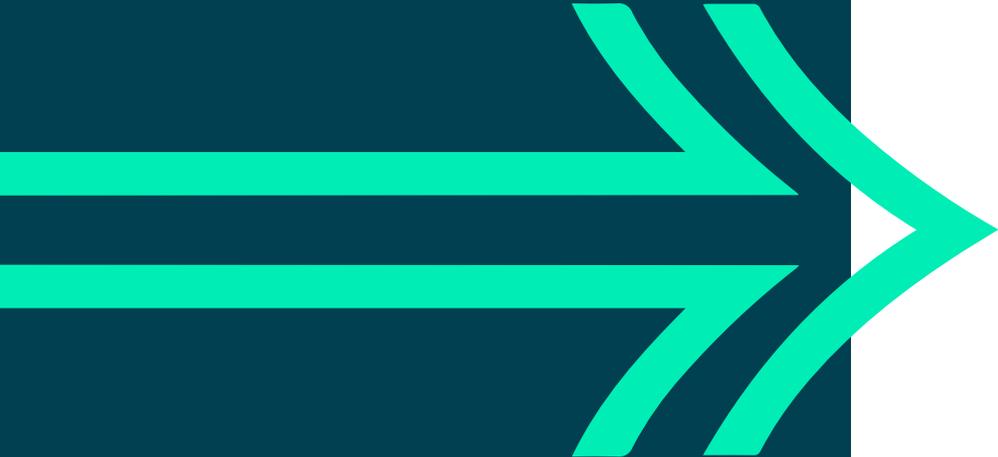
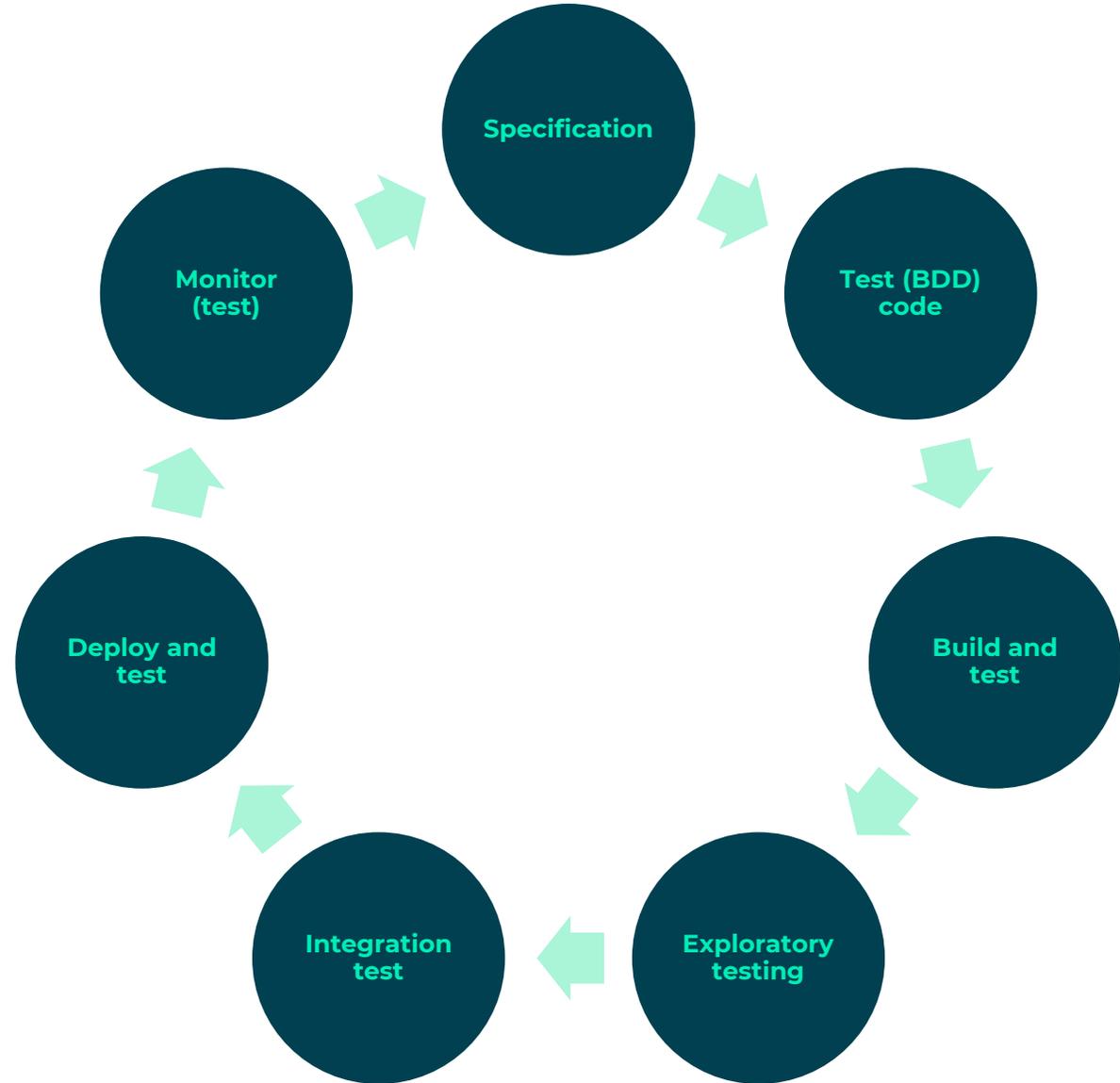
Continuous integration

Continuous delivery





TEST BASED DEVOPS

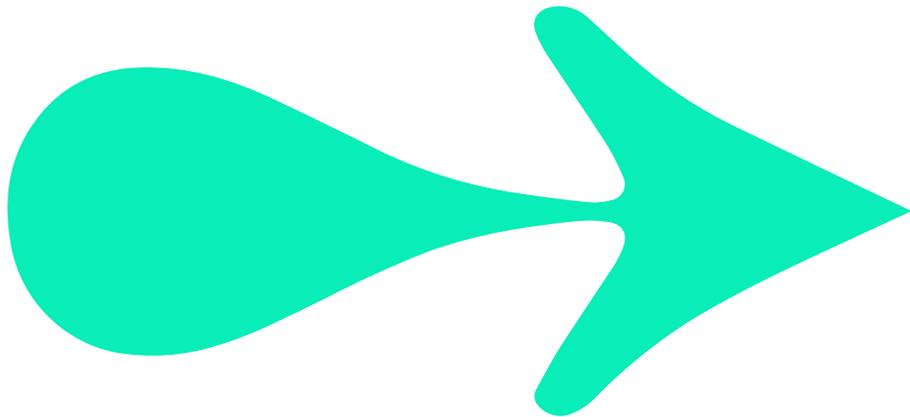




Schrodinger's Code

“Code that has not been written, but not tested, is in a state of neither working nor failing until it is observed.”

@SchrodingerCode



The Next Gen DevOps Lifecycle

The minimum an organisation evolving into DevOps practice should use is:

- A testing framework focused on behaviour
- An automated testing harness
- A strategy that focuses engineers on testing
- Applications designed from the bottom up to be testable
- Configuration management solutions that allow rapid creation of any environment
- Infrastructure that can be rapidly deployed
- Methods that can deploy products whenever they are ready
- Dashboards displaying the status of the build



The Next Gen DevOps Lifecycle

The minimum an organisation evolving into DevOps practice should use is:

- A testing framework focused on behaviour
- An automated testing harness
- A strategy that focuses engineers on testing
- Applications designed from the bottom up to be testable
- Configuration management solutions that allow rapid creation of any environment
- Infrastructure that can be rapidly deployed
- Methods that can deploy products whenever they are ready
- Dashboards displaying the status of the build



THANK YOU

Hope you enjoyed this learning journey.

