# Putting the Ops in DevOps

April Edwards
Senior Cloud Developer Advocate
DevOps Practice Lead

@TheAprilEdwards

@TheAprilEdwards


# Many thanks to our sponsors:

SquaredUp

au2mator

PATCH MY PC

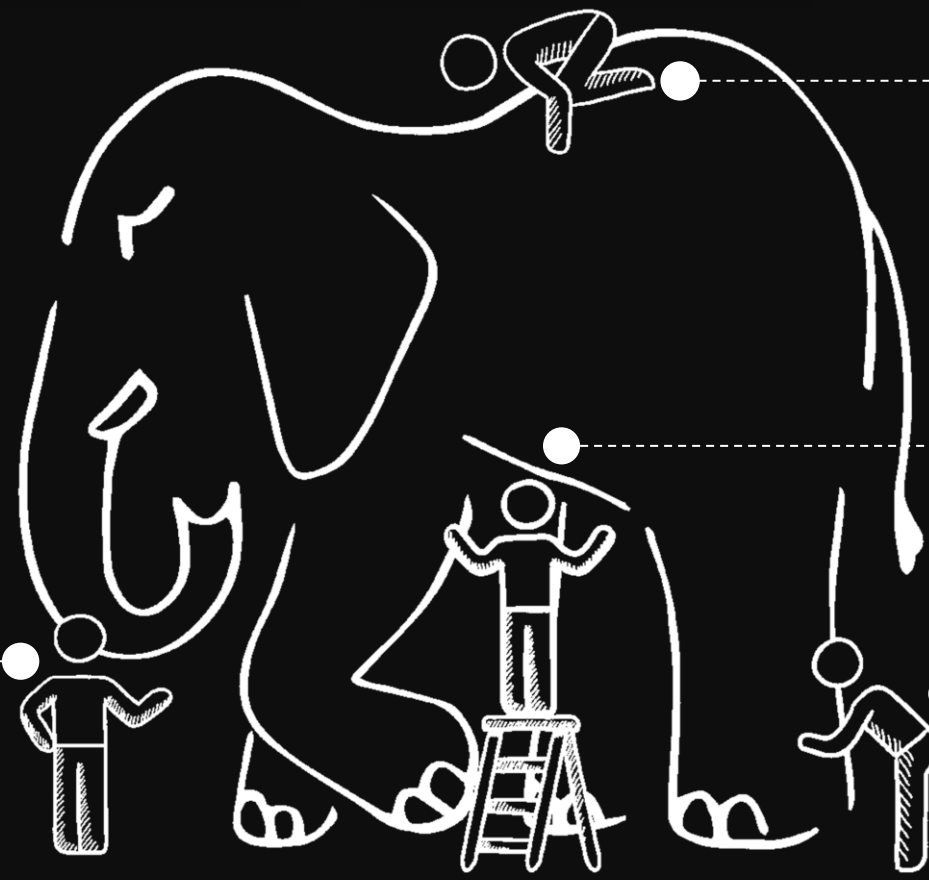
Chocolatey

System Frontier

ScriptRunner

@TheAprilEdwards

# What is DevOps?



It's a job title

It's automation

It means faster and smaller releases

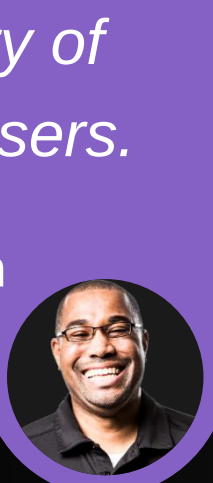
It's Development and Operations collaboration

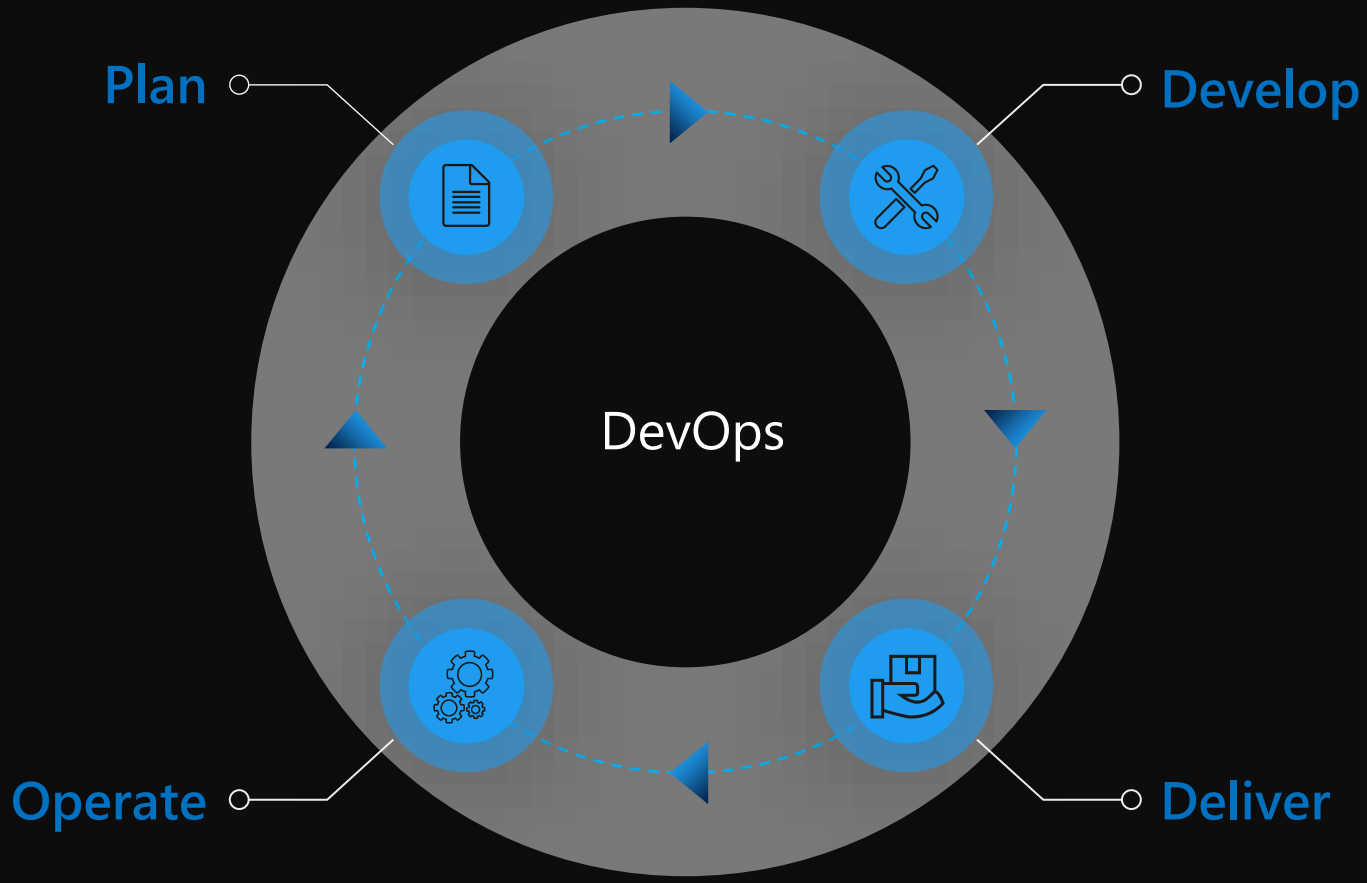
@TheAprilEdwards

Allianz

# What is DevOps?

> *DevOps is the union of **people**, **process**, and **products** to enable continuous delivery of value to your end users.*
>
> **– Donovan Brown**





# Why is DevOps important?

Your competition is already doing this

Increase velocity

Reduce downtime

Reduce human error

# High performance DevOps companies achieve...

973X more frequent deployments

6570X faster from commit to deploy

**Faster time-to-market**

**Increased revenue**

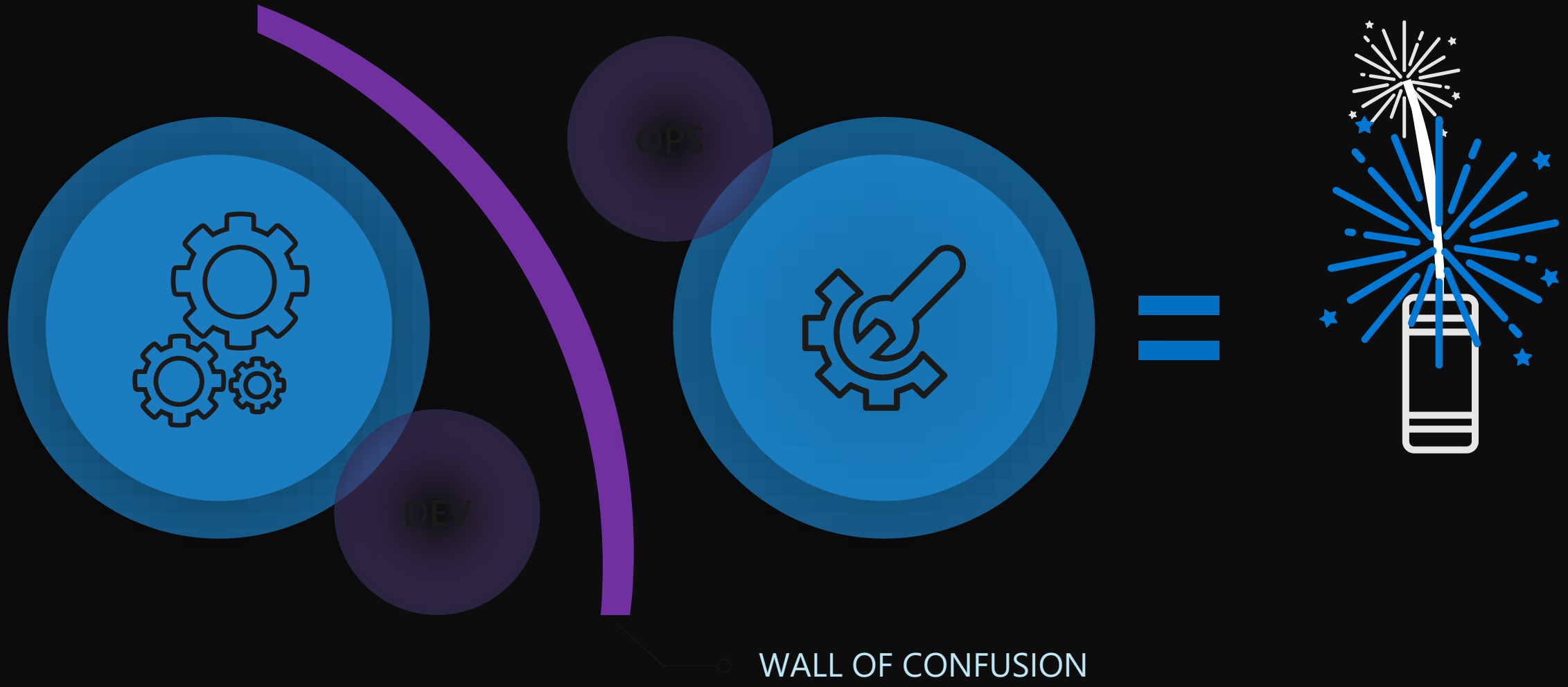
3X lower change failure rate

6570X faster Incident recovery

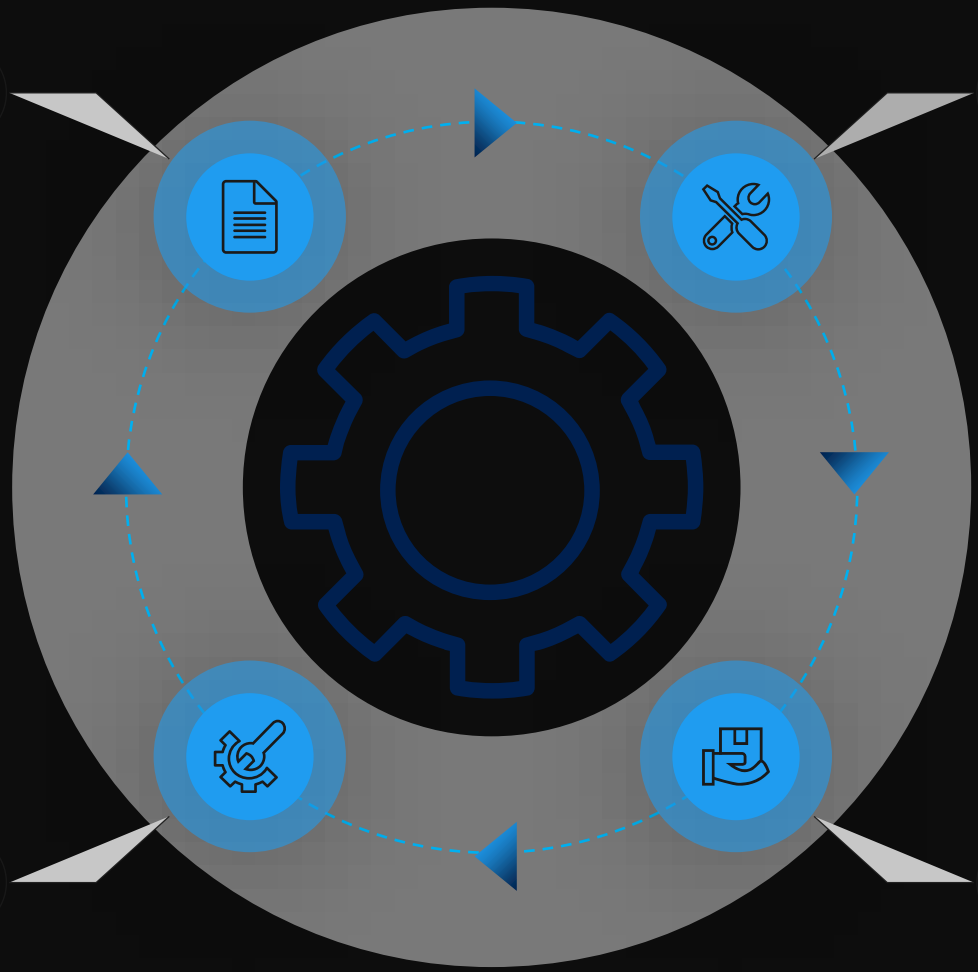

Source: 2021 Accelerate State of DevOps Report

@TheAprilEdwards

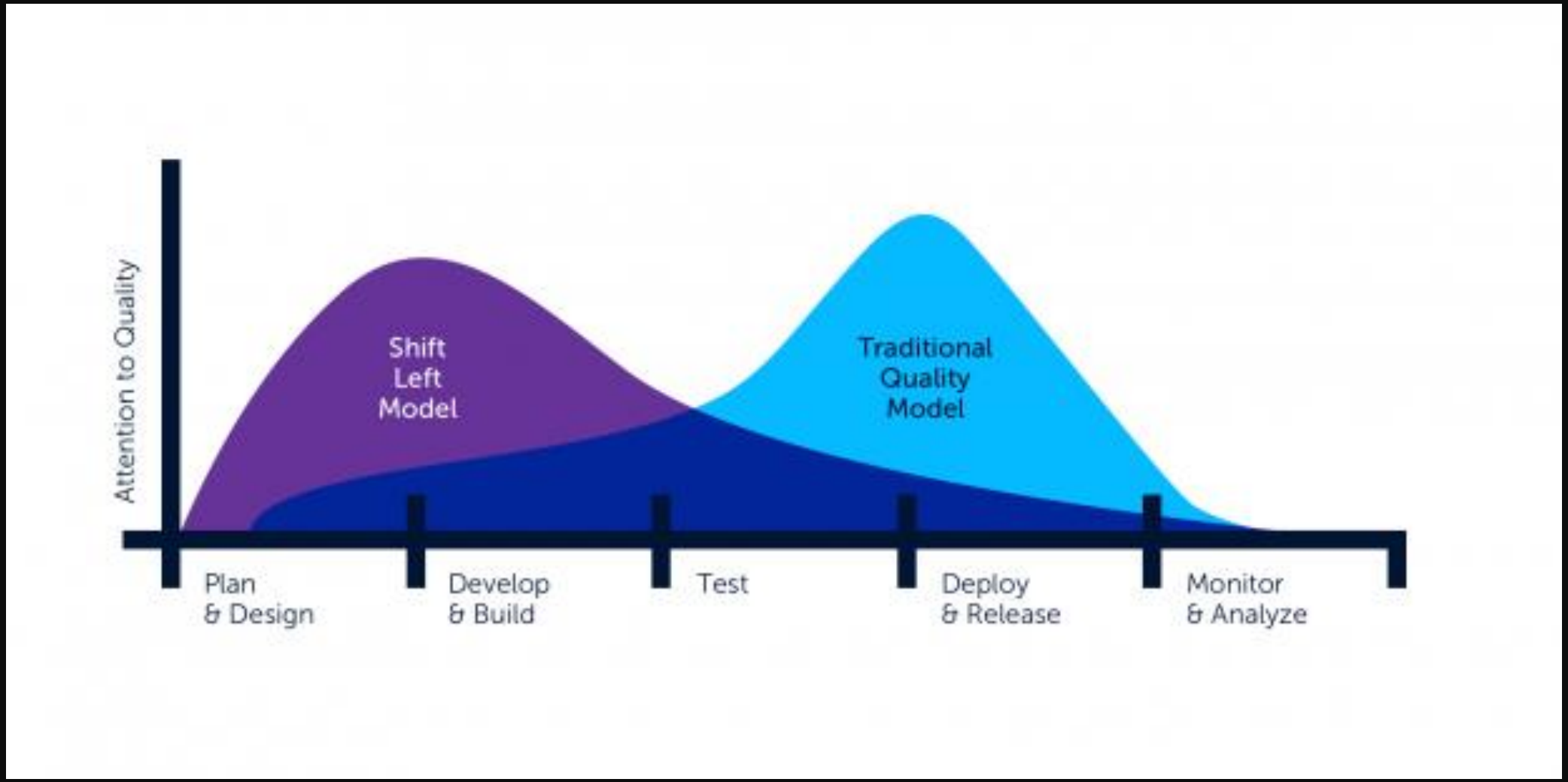
# People



# Process


Development
Plan 1
2 Develop
Operate 4
3 Deliver
Production

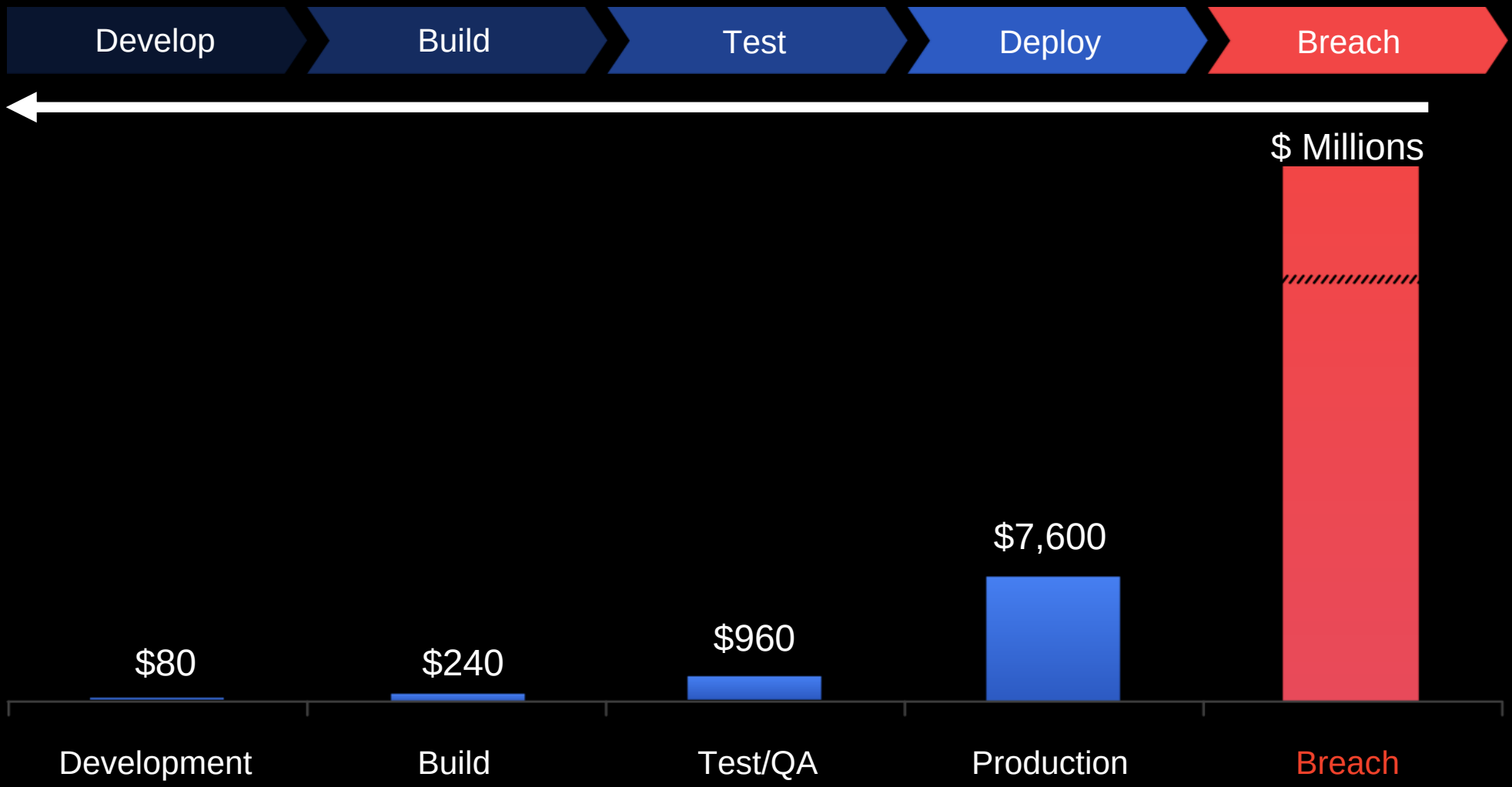
# Shift Left






@TheAprilEdwards

# Sources of vulnerabilities




Sources: NIST, Polemon Institute

# Continuous delivery

Deploy often

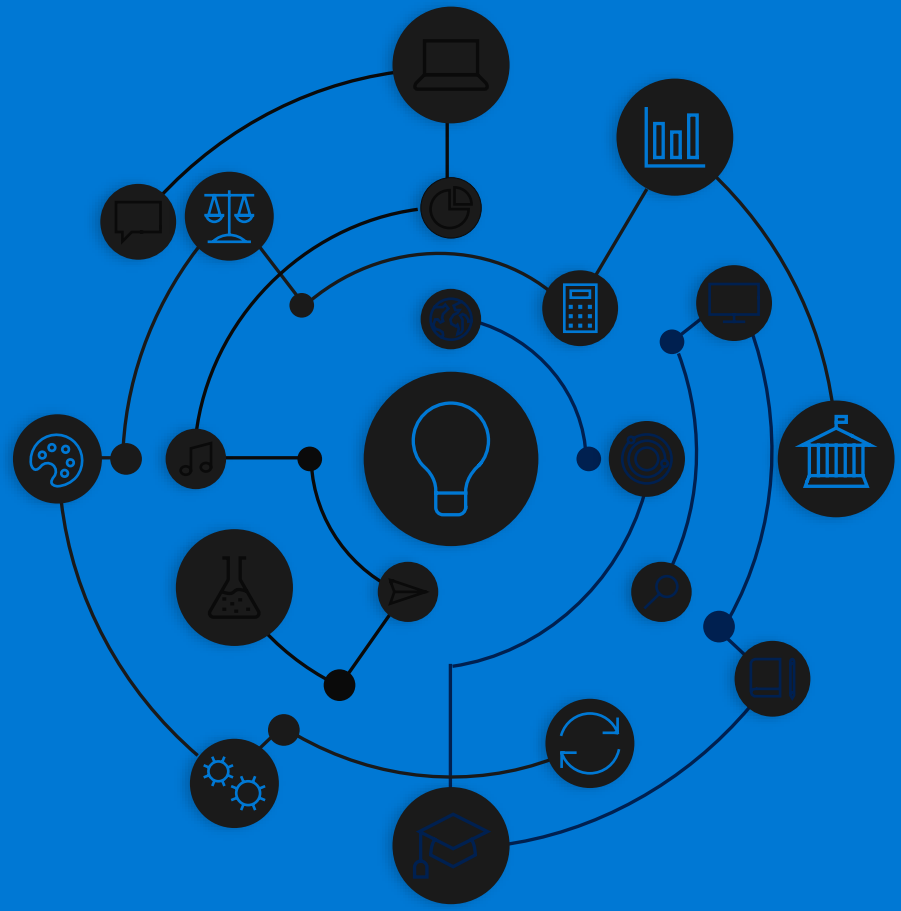
Deploy smaller / focused changes

Initiated by code commit

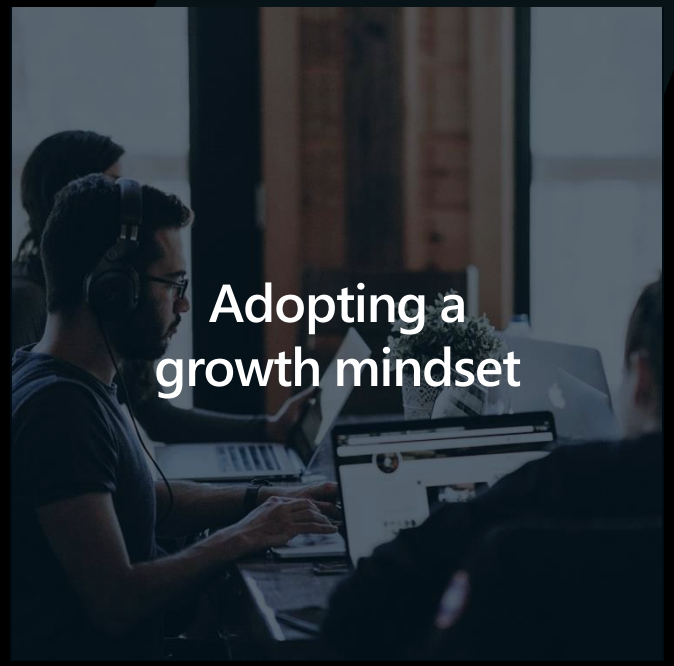
Automated testing

Feedback engines

Historical records



# Key Elements of Microsoft's DevOps Culture





# What we've done throughout our DevOps journey

## Changed our definition of done

We rely on the Idea to Data (I2D) method, which involves updating our key performance indicators (KPIs) and collecting telemetry to continually measure the value our products bring customers.

## Adopted a production-first mindset

Engineering teams own the entire lifecycle of a feature, from inception through operation. As issues or incidents arise, engineering teams are responsible for owning the customer response.

# What we've done throughout our DevOps journey (cont'd)

## Changed how we collaborate, develop, and deliver

We restructure to extend the management lifecycle for developers beyond version release. We teach the people who build solutions to be responsible for the operation, fixes, troubleshooting and ownership of each line of code they write.

## Enhanced security from dev to production

We use zero-trust as a model to help protect our infrastructure through enforced device health, strong authentication, least-privileged access, and pervasive telemetry that verifies control effectiveness.

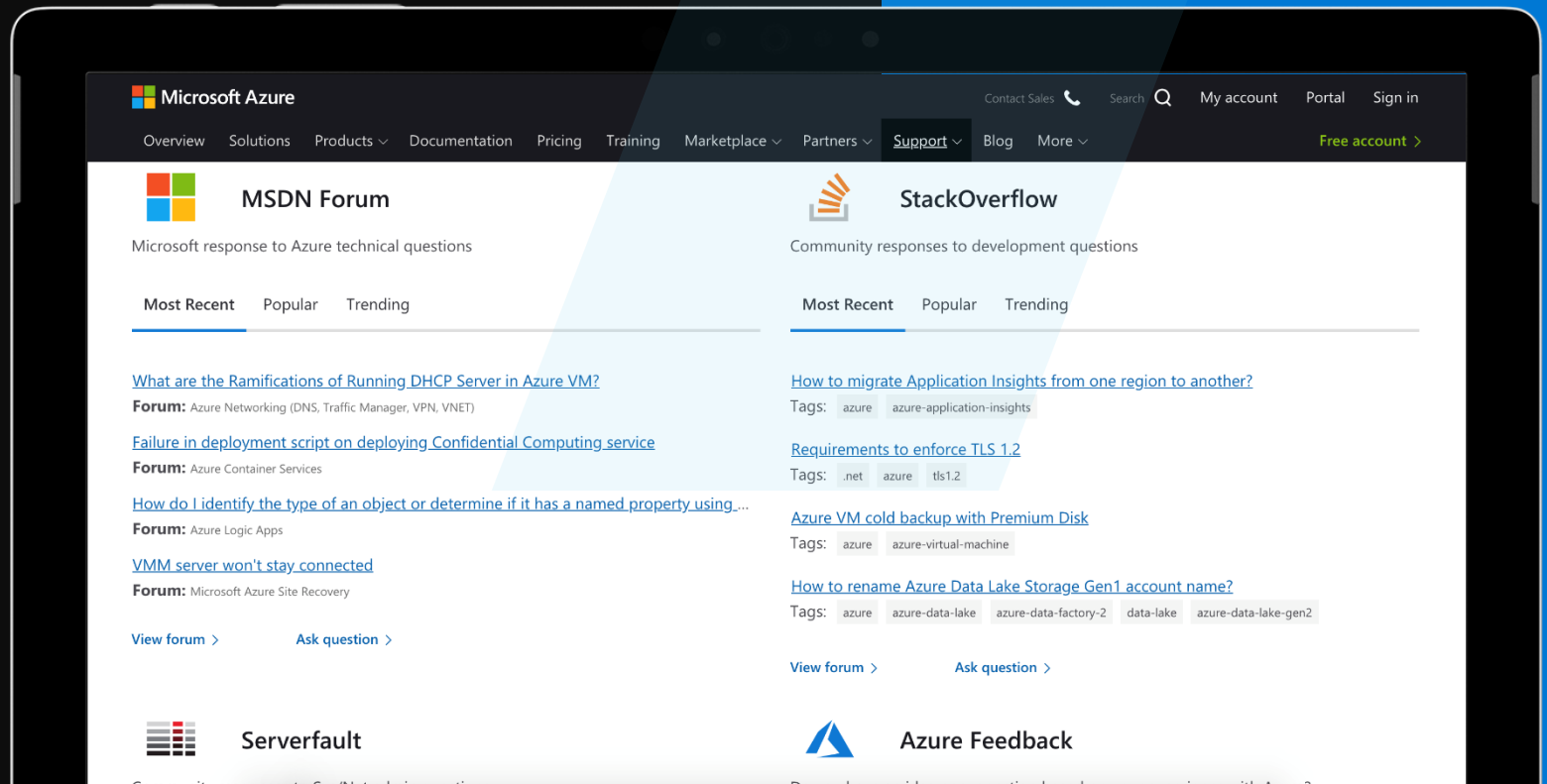
Change your definition of done

# Take a zero-distance approach by listening to your customers

Gathering customer feedback early and often is key to building empathy between engineers and customers.

## Forums and social

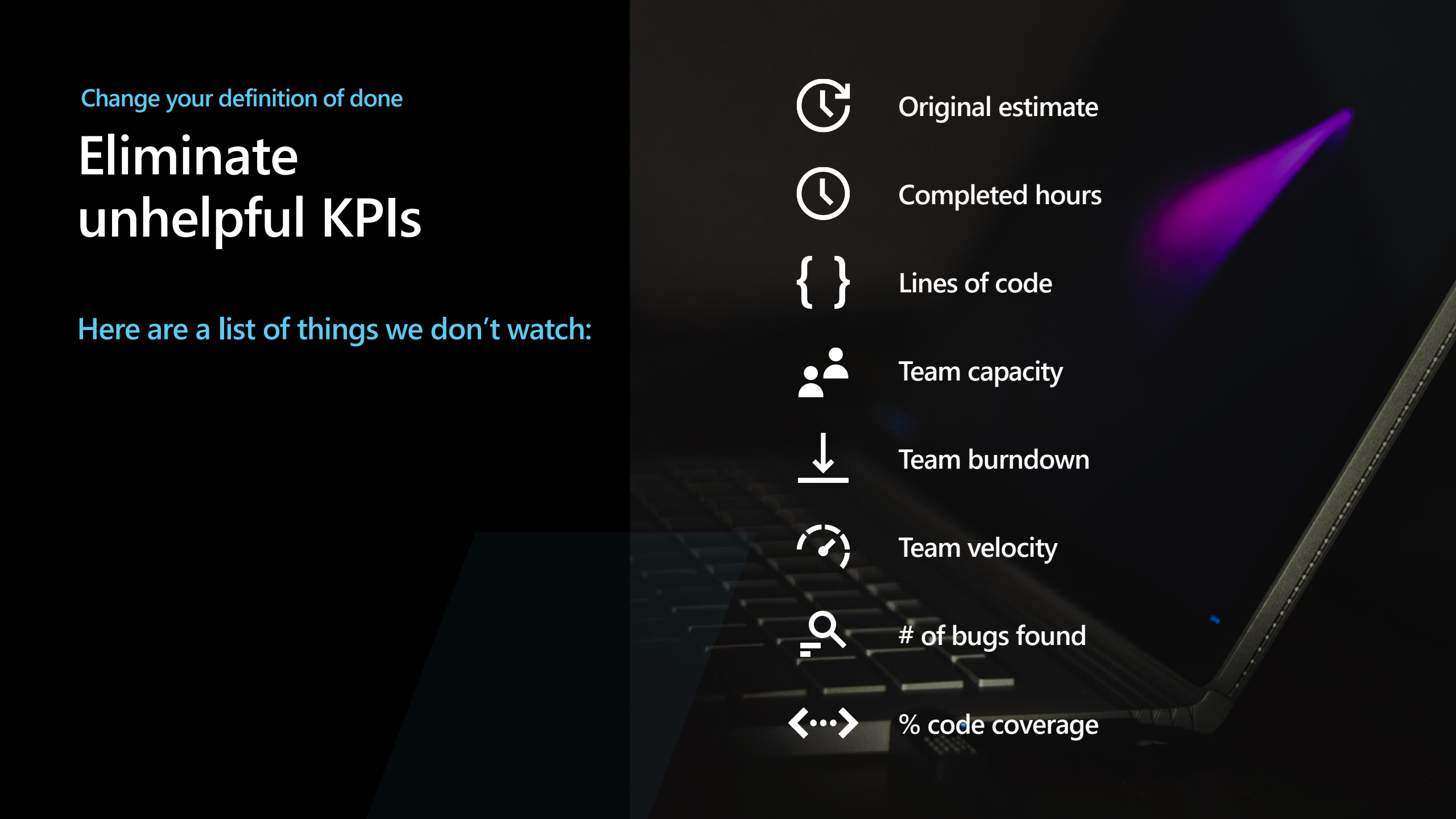
Automatically feed data into the work item tracking system



Change your definition of done

# Eliminate unhelpful KPIs

Here are a list of things we don't watch:



Change your definition of done

# Change what you track

## Focus on measuring only the most critical and impactful KPIs:

### Customer usage

How much value are users getting?

- Acquisition
- Retention
- Engagement
- Satisfaction (NPS)
- Feature usage

### Pipeline throughput

How efficient is the DevOps process?

- Time to build
- Time to test
- Time to deploy
- Time to improve
- Failed and flaky automation

### Live site health

How quickly can you detect and fix issues?

- Time to detect, time to communicate, time to mitigate
- Customer impact, customer support metrics
- Incident prevention items
- Aging live site problems
- SLA per customer

Adopt a product-centric mindset

# Measure and improve

## Incident response by month

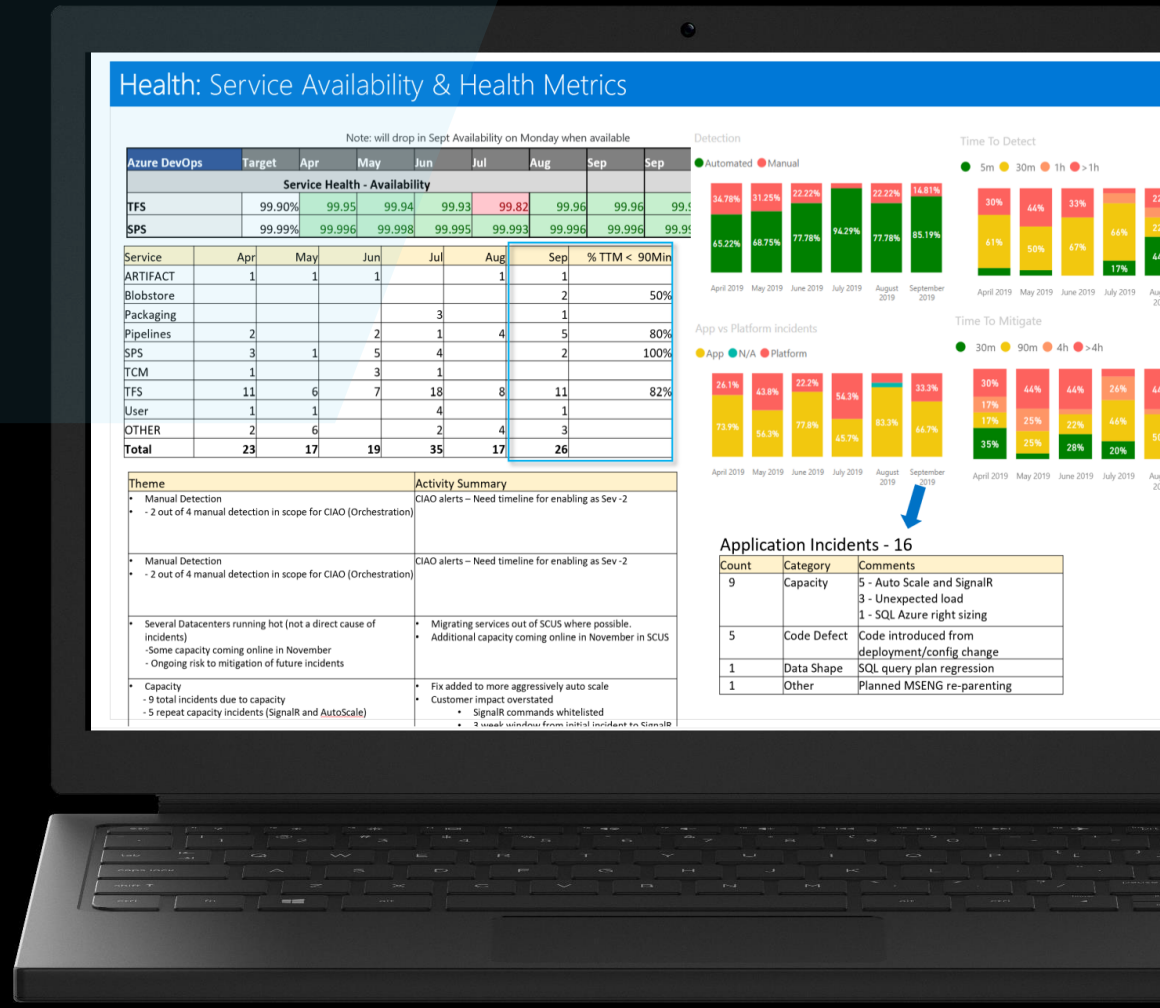
- Availability (per customer)
- Automated detection
- Time to detect
- Time to mitigate

## Every incident that affected users

- Identify patterns
- Ensure remediation

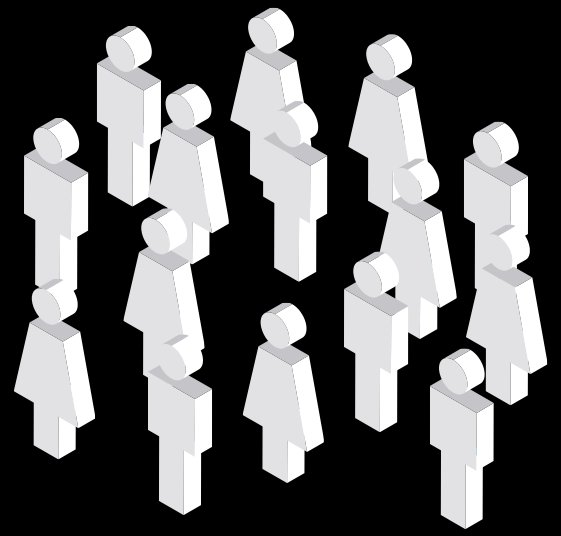
## All customer support inquiries

## Cost of operation

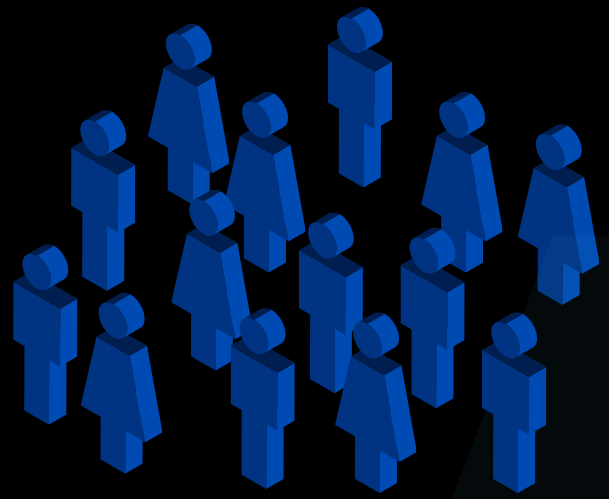


Adopt a product-centric mindset

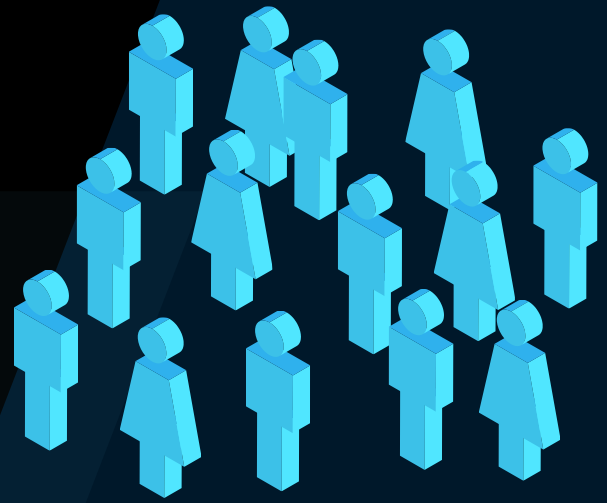
# Evolve the organization



Program management



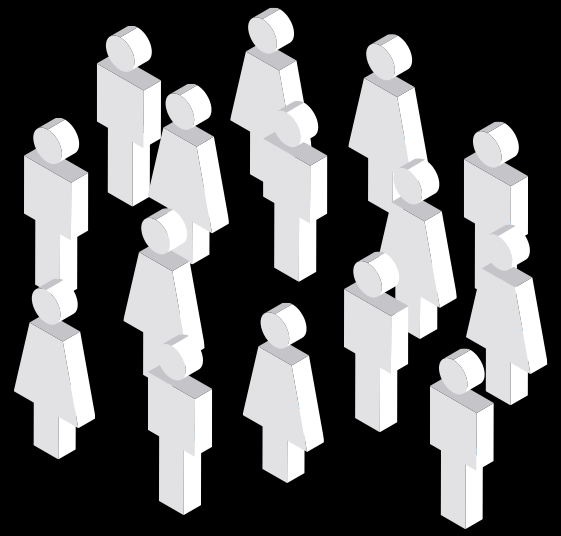
Development



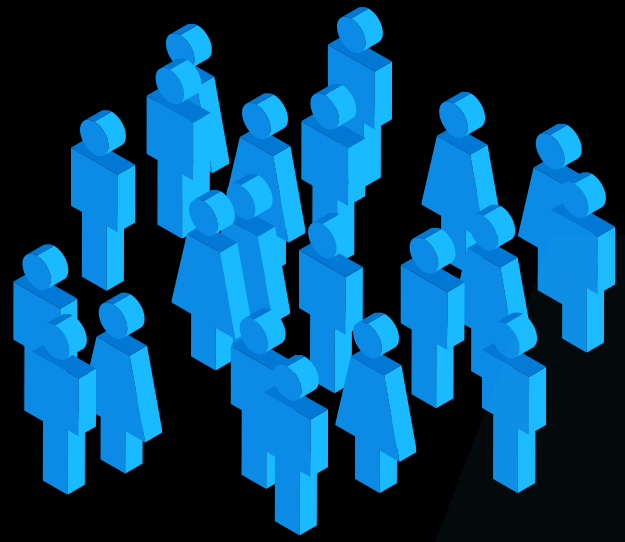
Testing

Adopt a product-centric mindset

# Evolve the organization



Program management

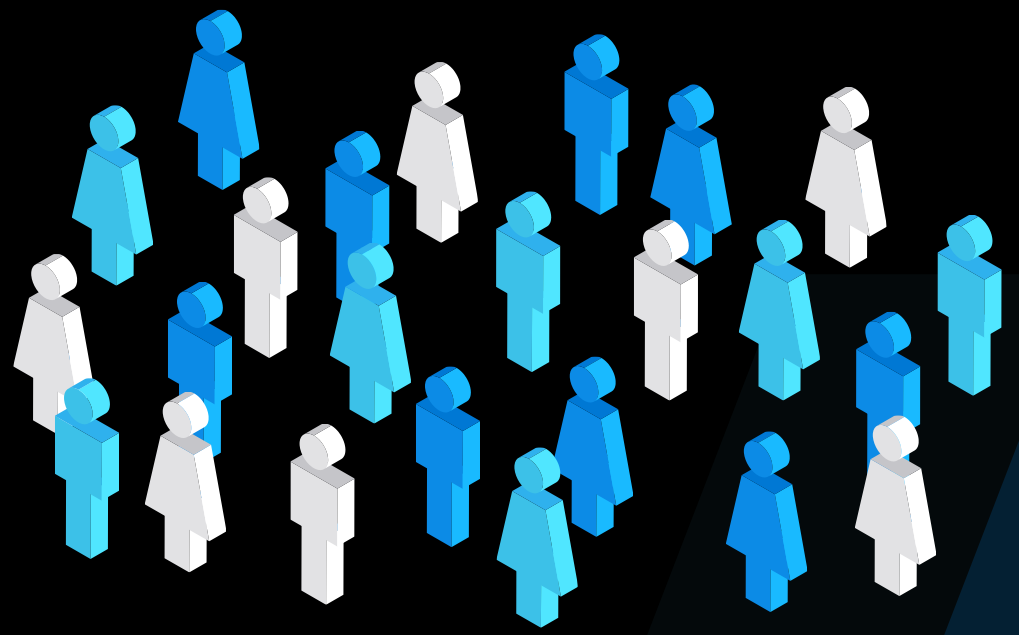


Engineering

Ops/SRE

Adopt a product-centric mindset

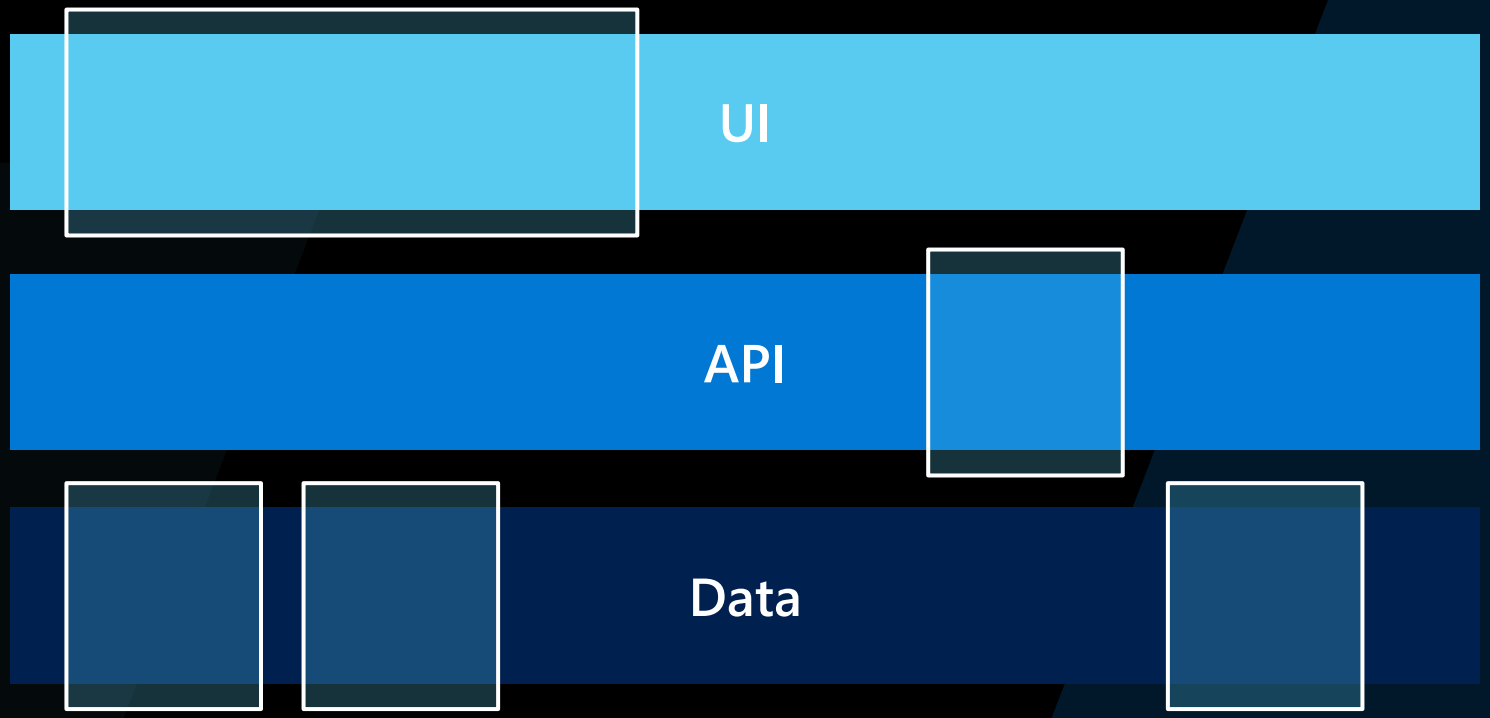
# Evolve the organization



Feature team

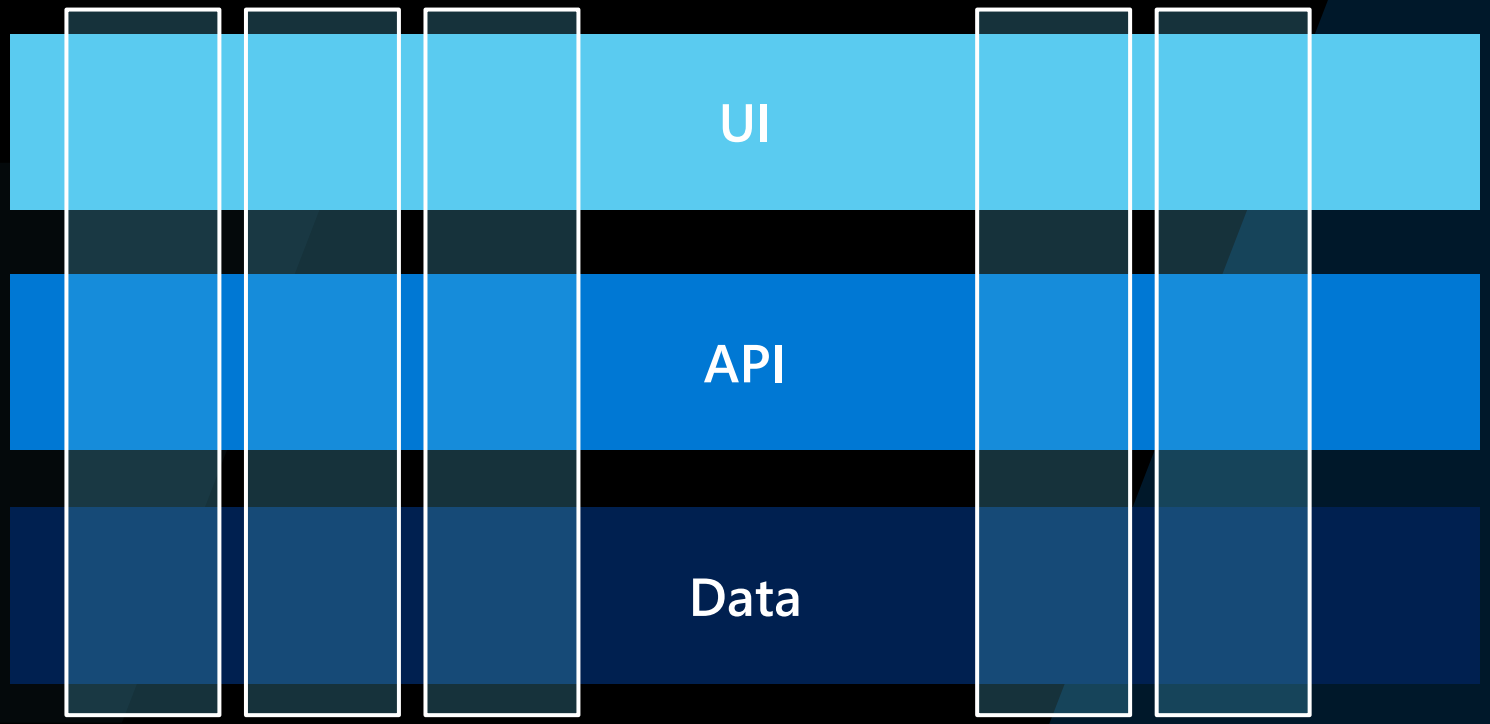
Adopt a product-centric mindset

# Instead of horizontal…



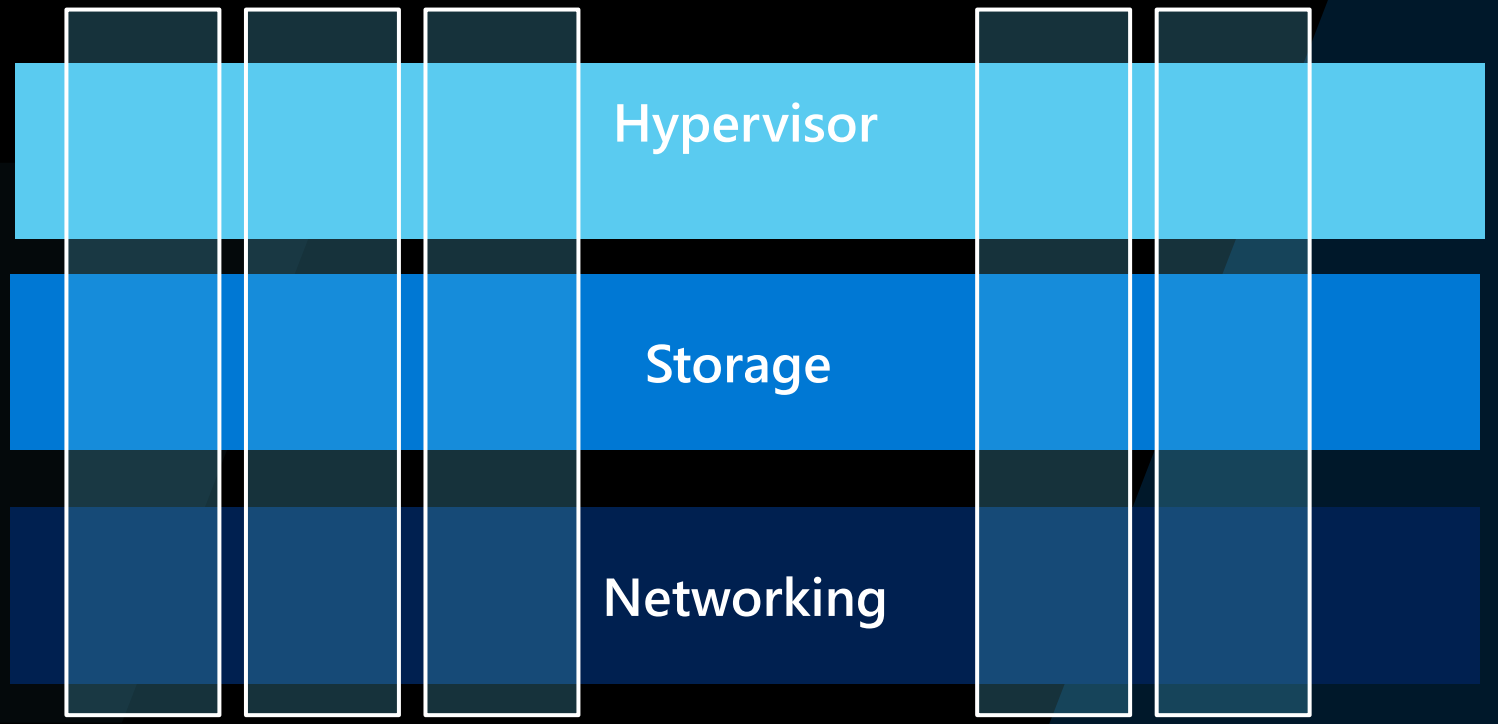
Adopt a product-centric mindset

# We strive for vertical



Adopt a product-centric mindset

# Strive for vertical operations



Adopt a product-centric mindset

# Evolution of full-stack teams

Physical team rooms (augmenting remote workers)

Cross discipline

1 Engineering Lead + 1 Product Owner

10–12 engineers

Self managing

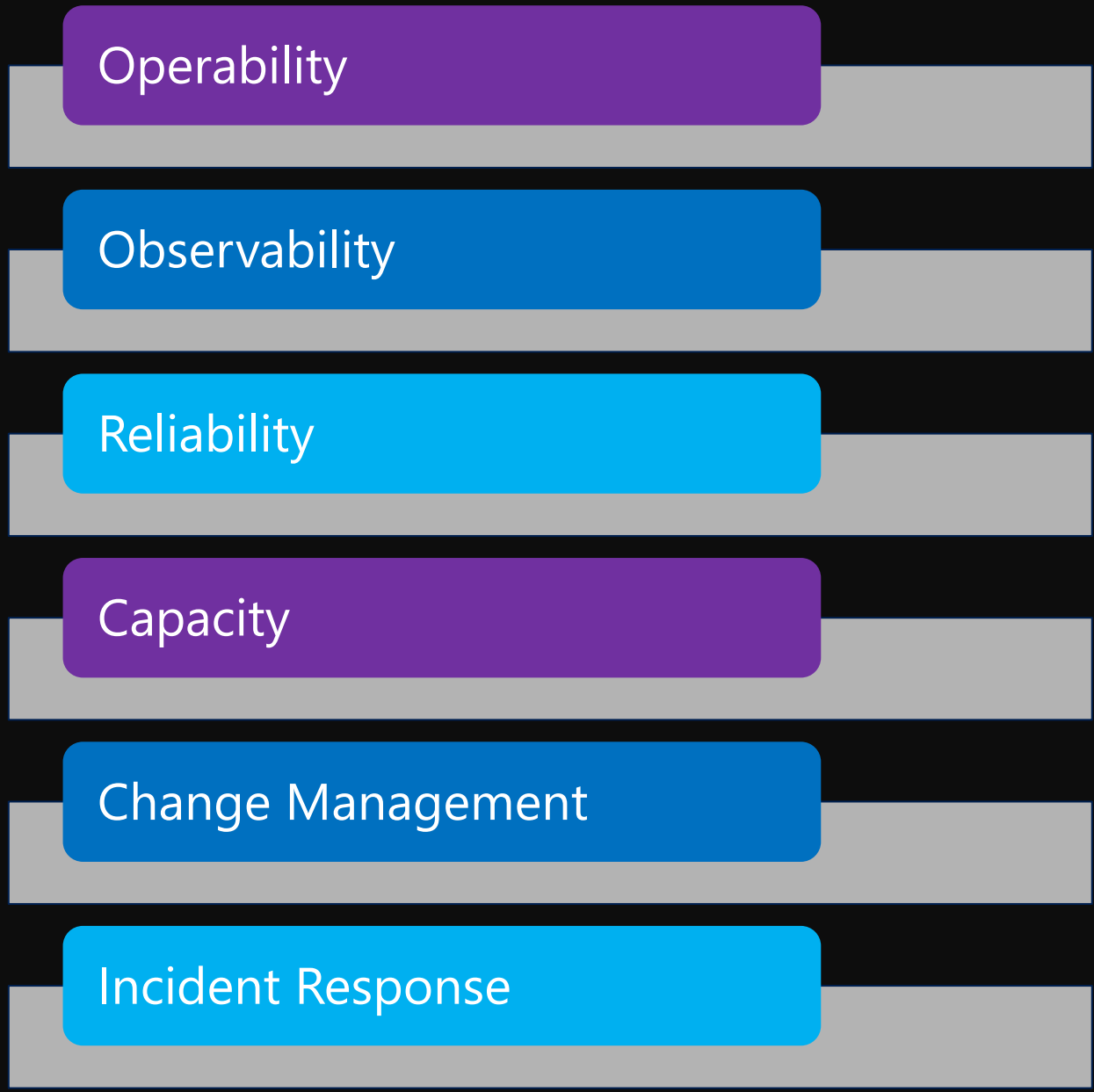
Clear charter and goals based on OKR's

Intact for 12–18 months

Own features in production

Own deployment of features

# Pillars of Modern Operations



# You Get What You Measure

(Don’t Measure What You Don’t Want)

@TheAprilEdwards

# The Time Thieves

Unknown dependencies

Too much work (WIP)

Conflicting Priorities

Unplanned work

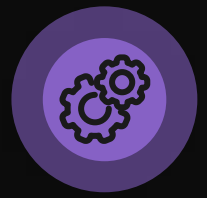
Neglected work

# Reducing Manual Intervention in Production

Move manual, repetitive tasks to automation

Minimize long term user principal role assignments

Use a central secret store



# Automation to the rescue!

# Tooling over Tasks

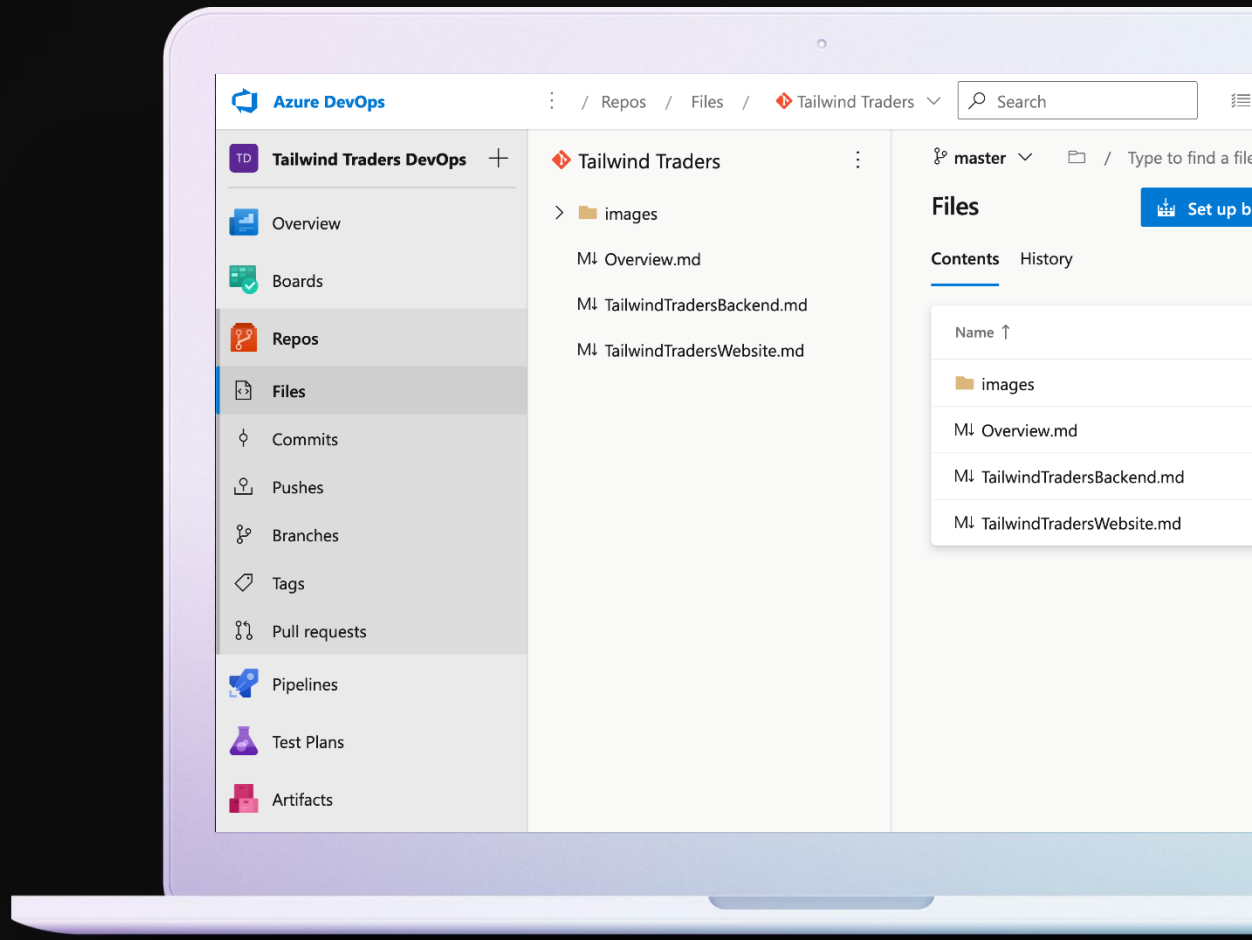
# Source Control

*Introduction to Git and GitHub*

@TheAprilEdwards

# What is Source Control and Why do we Need it?

- A form of version control
- Uses concept of code repositories
- Tracks changes made within repositories
- Allows for cross-team collaboration
- GitHub




# GitHub

The largest developer community on the planet

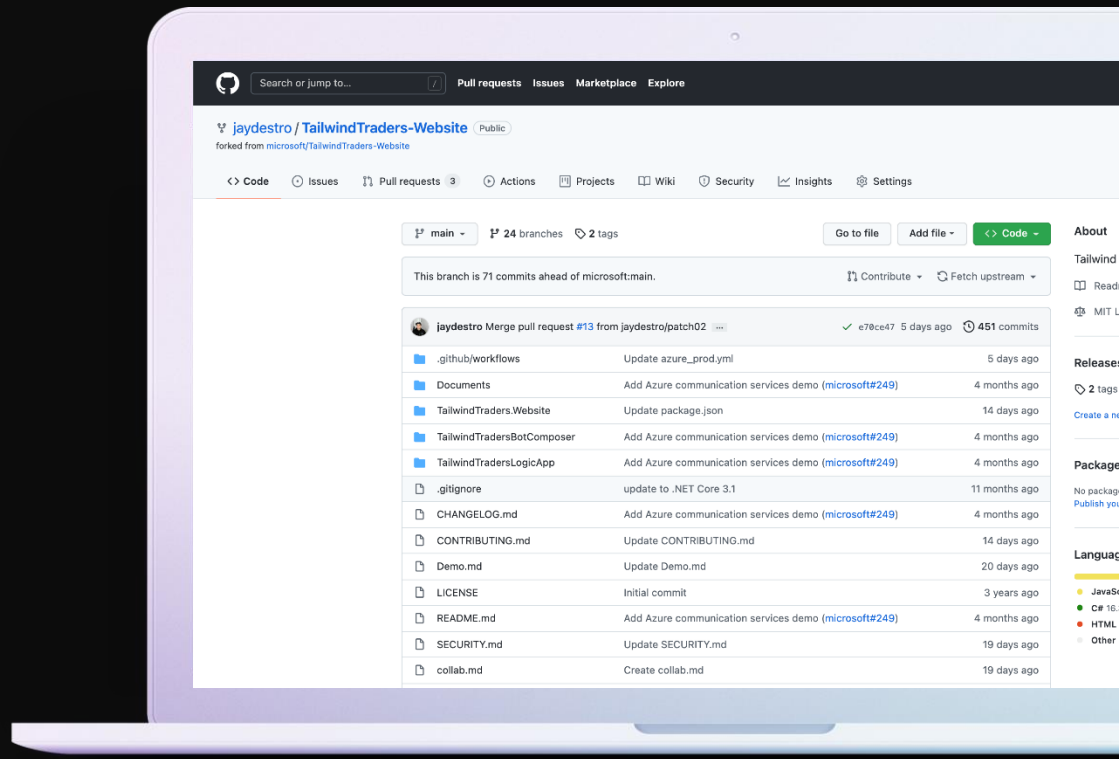


@TheAprilEdwards

# What is GitHub?

GitHub is the leader in Git repository hosting. Some key features of GitHub

- Expertise sharing
- Cross-team collaboration
- Improved code reuse
- Codespaces on GitHub
- GitHub Actions (CI/CD)
- Increased velocity

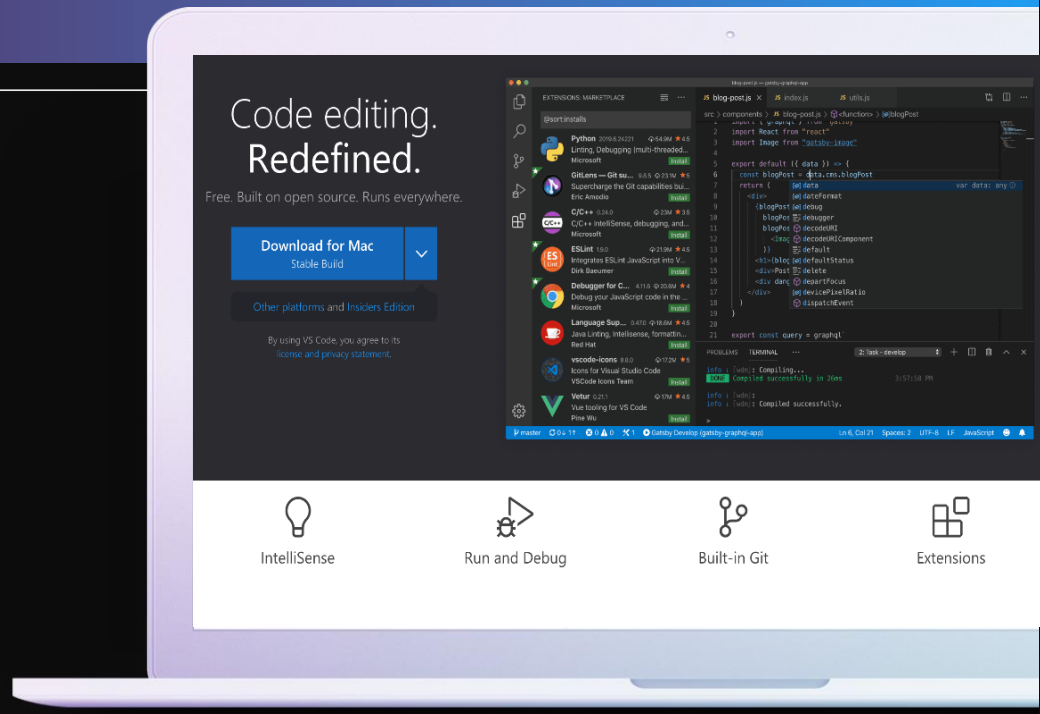



@TheAprilEdwards

# What is Visual Studio Code?

Visual Studio Code is a lightweight and powerful source code editor.

- Run anywhere (Mac, Win, Lin)
- Git commands built-in
- Extensible and customizable
- IntelliSense syntax highlights
- Easily debug code
- Open Source
- Free!




@TheAprilEdwards

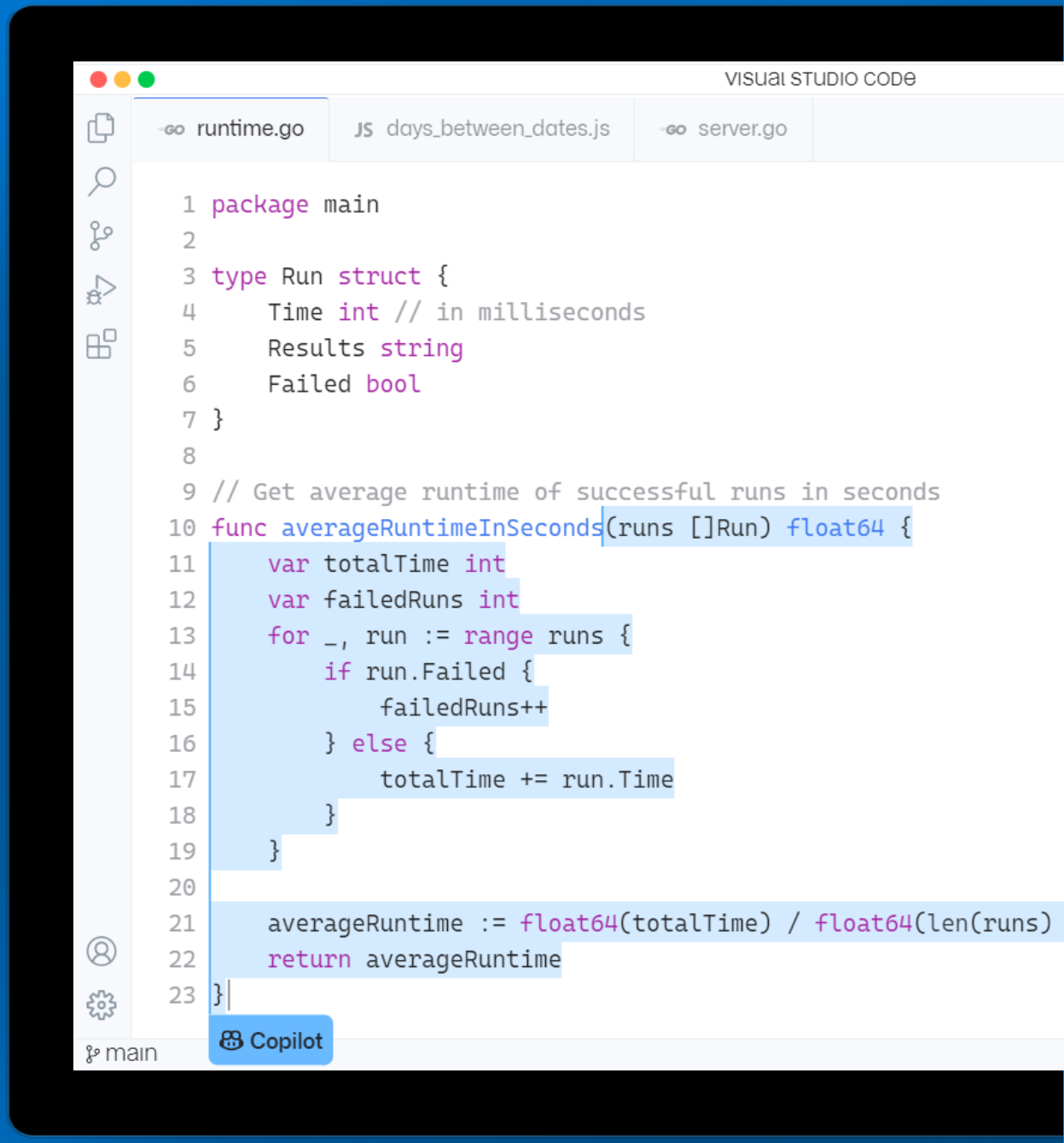
# GitHub Copilot

Your AI pair programmer

What is GitHub Copilot?

Convert comments to code.

Tests without the toil.



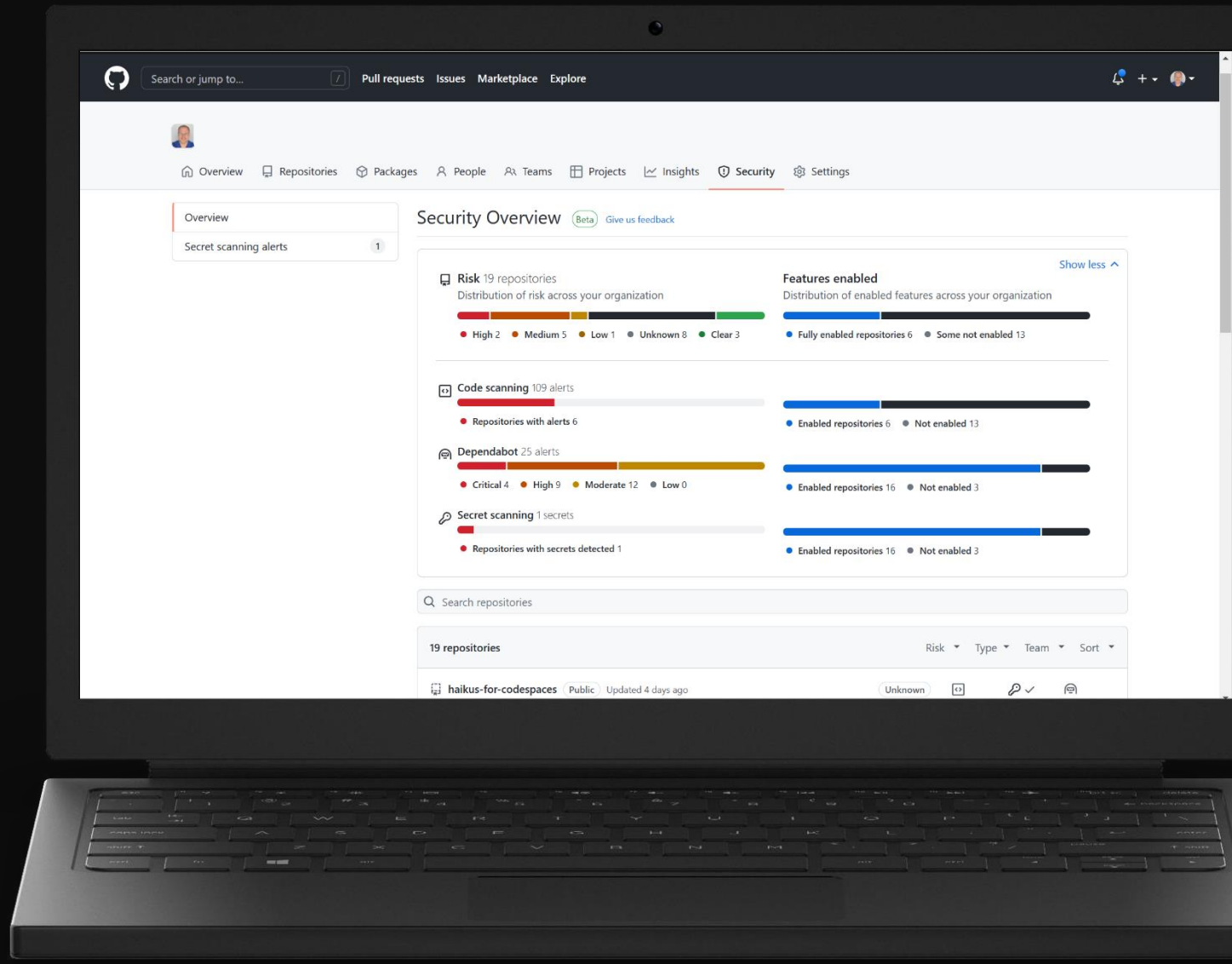
# Code security

Secure the software supply chain

Secure the usage of open source

Code Scanning

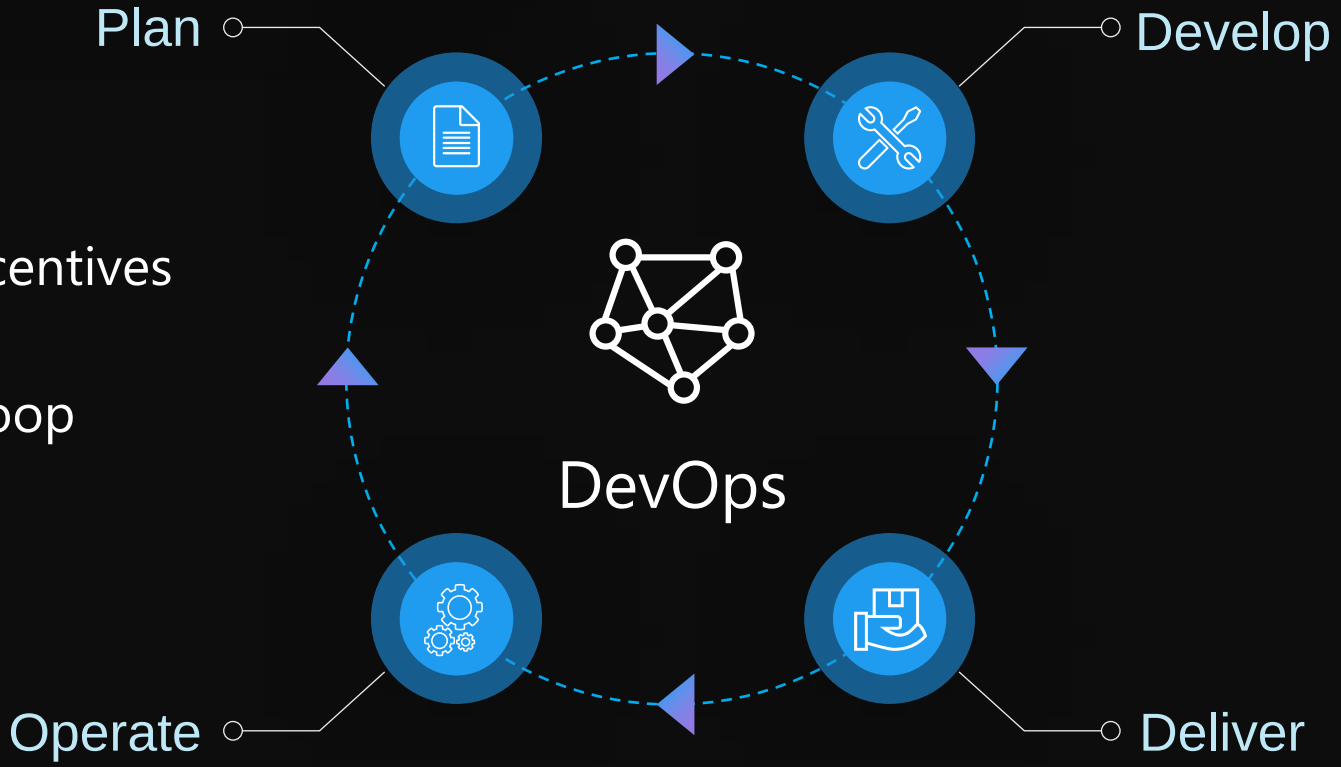
Secret Scanning

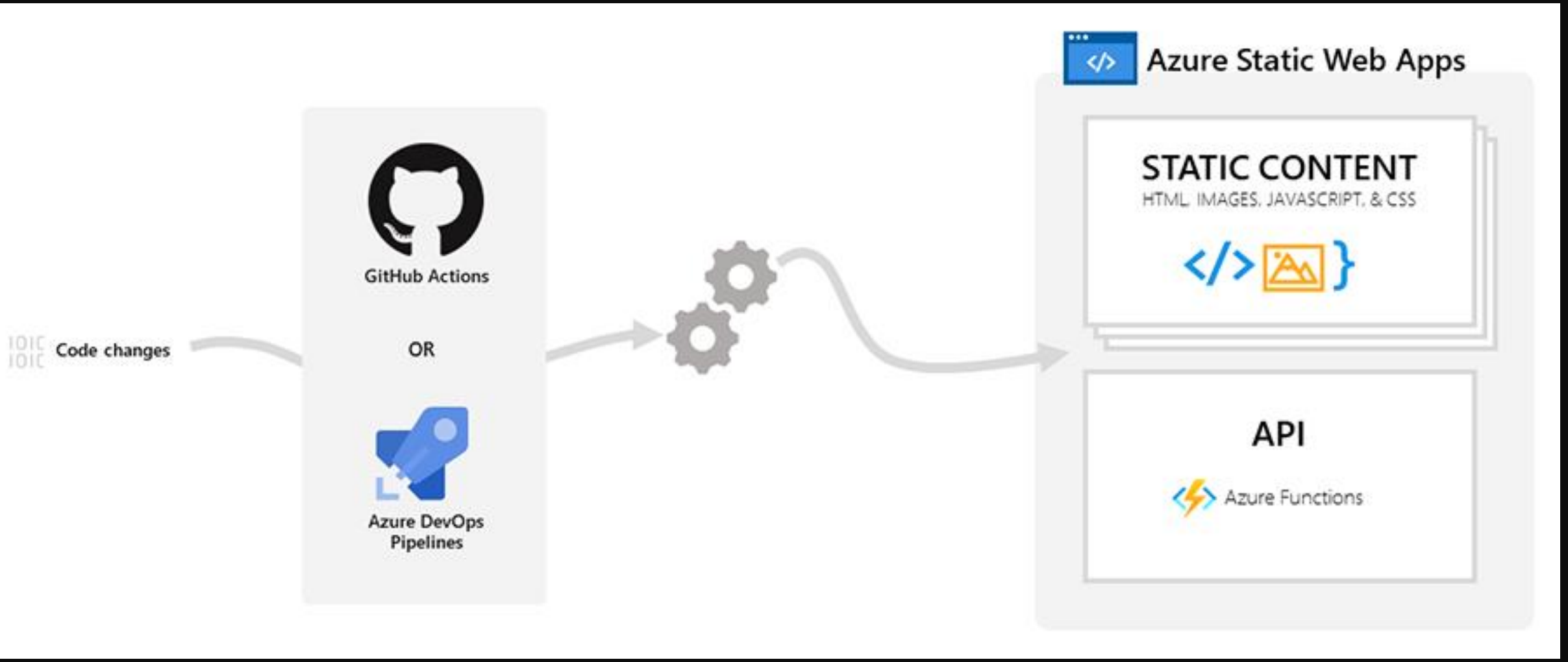



@TheAprilEdwards

# What does this all mean?

- Deliver value
- Increasing efficiency
- Alignment of Goals and Incentives
- Streamline the feedback loop
- Continuously improve
- Deliver faster
- Autonomy








@TheAprilEdwards

# Resources

The DevOps Lab: aka.ms/TheDevOpsLab

Microsoft Learn: aka.ms/MSLearnUK

DevOps Blog: aka.ms/devopsblog

NubesGen: Nubesgen.com

# Thank You !

@TheAprilEdwards