DevOps Fundamentals

What is DevOps

Agenda

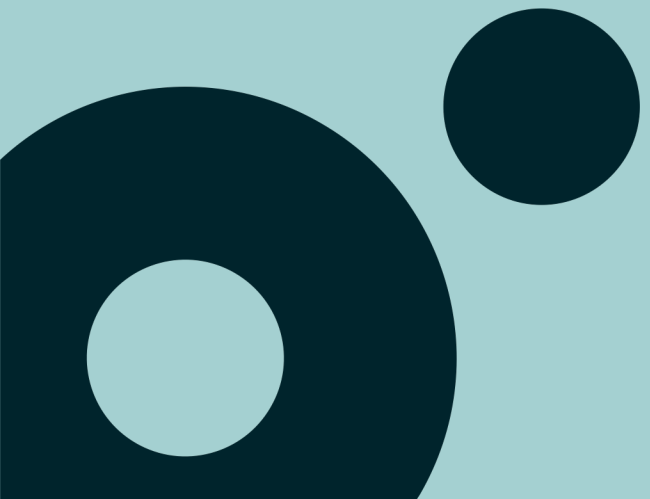
What is DevOps?

DevOps Journey

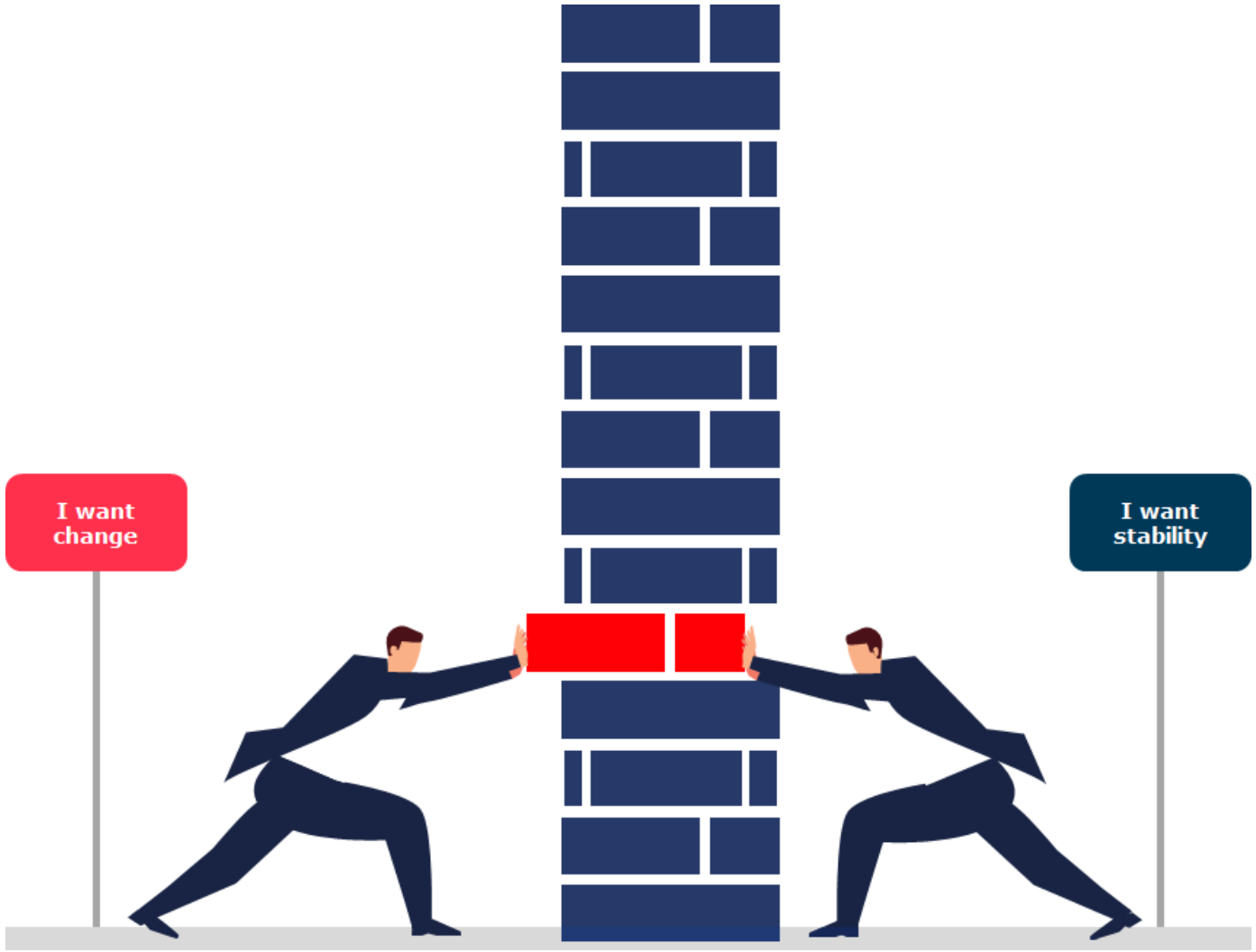
DevOps Challenges

moOngy.

What is DevOps?



The Wall of Confusion



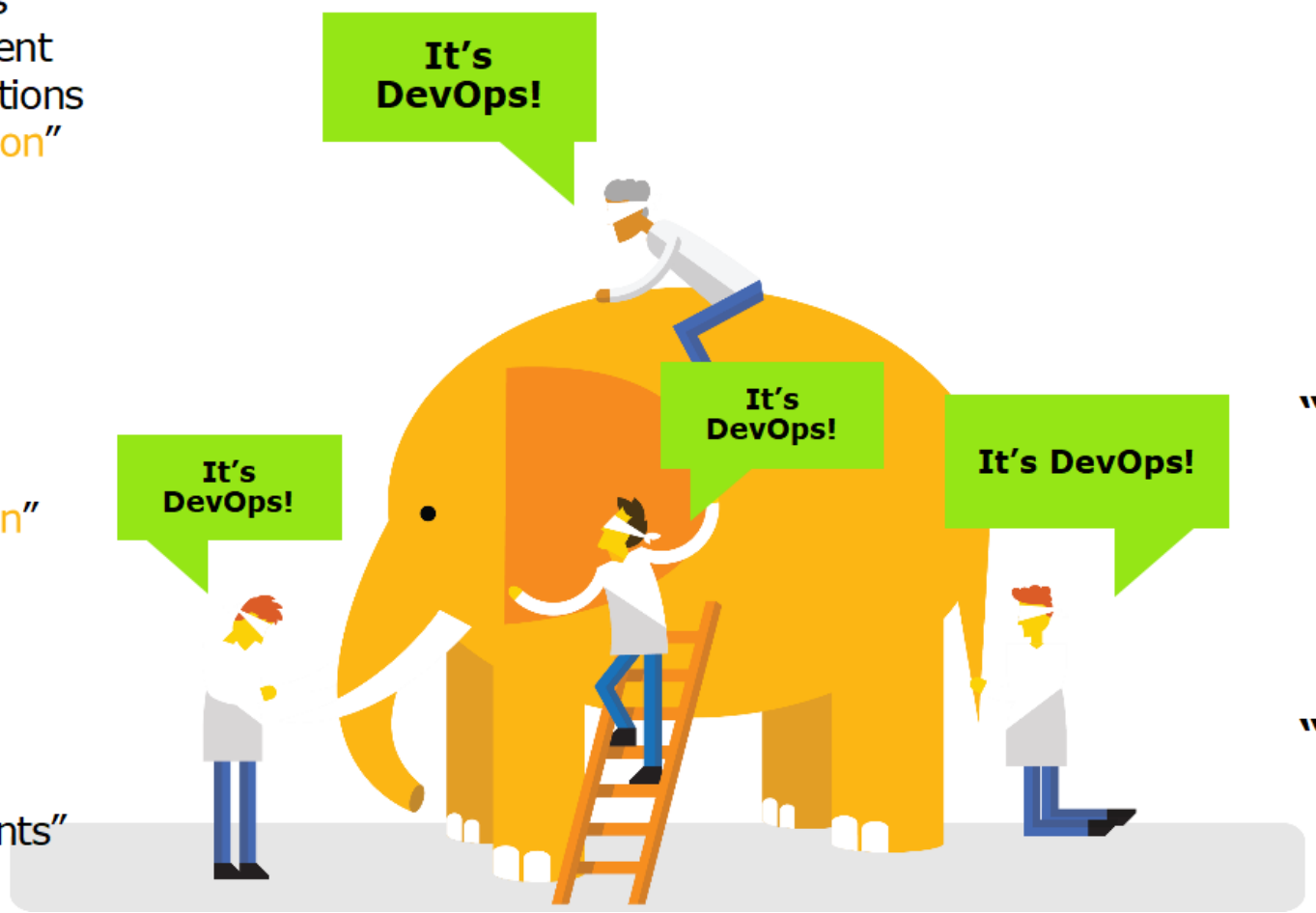
Unite Dev and Ops: Breaking the Silos

Definition of DevOps

"**DevOps** is development and operations collaboration"

"**DevOps** is using automation"

"**DevOps** is small deployments"



"**DevOps** is treating your infrastructure as code"

"**DevOps** is feature switches"

"**DevOps** is continuous monitoring"

moOngy.

Definition of DevOps

"DevOps is a set of practices that combines software development (Dev) and IT operations (Ops). It aims to shorten the systems development life cycle and provide continuous delivery with high software quality"

By Wikipedia

"DevOps is the union of people, process, and product to enable continuous delivery of value to your end users."

By Microsoft

"DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity."

By Amazon

"Architectural practices, technical practices, and cultural norms that allow us to increase our ability to deliver applications and services quickly and safely."

By Gene Kim

Definition of DevOps

"DevOps is a set of practices that combines software development (Dev) and IT operations (Ops). It aims to shorten the systems development life cycle and provide continuous delivery with high software quality"

By Wikipedia

"DevOps is the union of people, process, and product to enable continuous delivery of value to your end users."

By Microsoft

"DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity."

By Amazon

"Architectural practices, technical practices, and cultural norms that allow us to increase our ability to deliver applications and services quickly and safely."

By Gene Kim

Definition of DevOps

Focus on continually deliver value

What is value?

- Understand your end user
- Less is more
- Focus on velocity but always with quality

Having your teams working together (Break the silos)

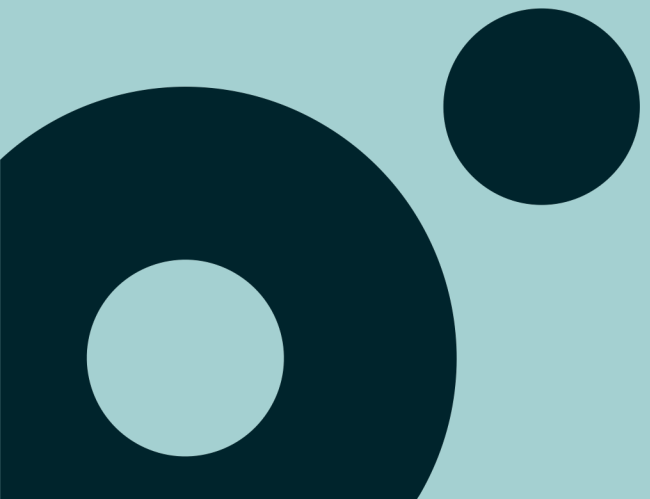
Create multi-skilled teams (don't mean multi-skilled people)

Automate everything you can

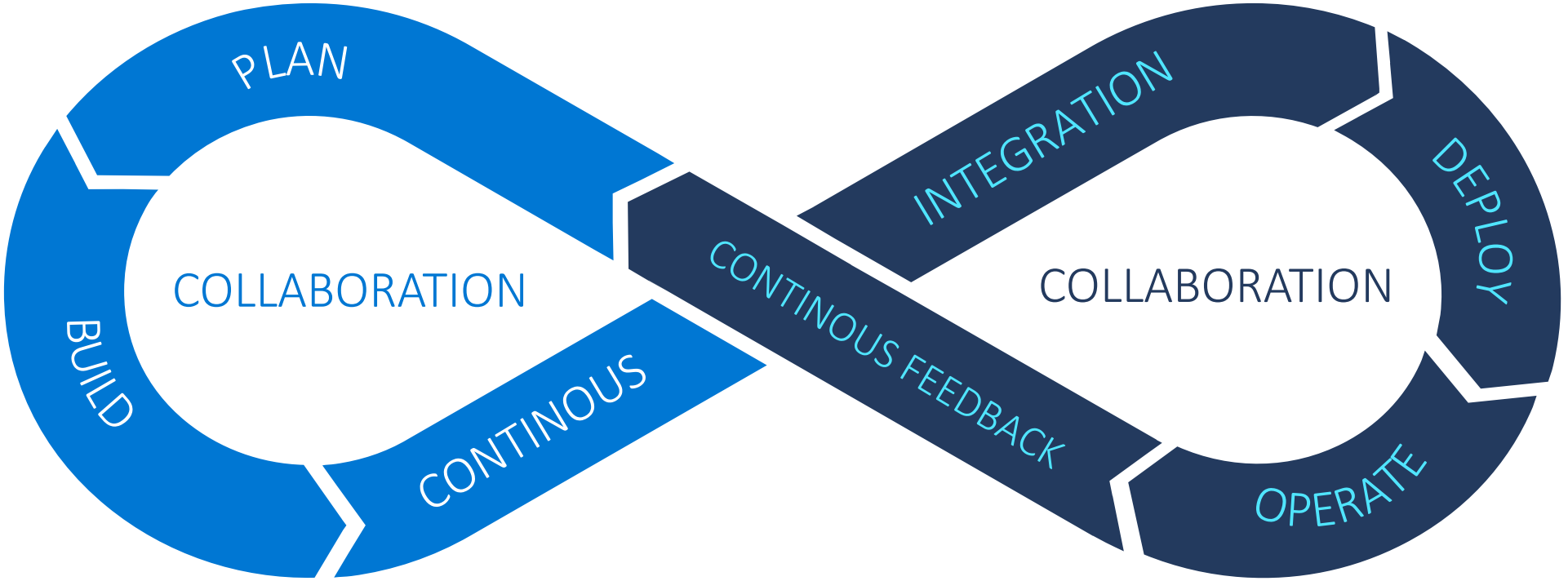
Autonomy and Accounting

Always improve!

DevOps Journey



DevOps Journey: Infinite Loop



DevOps Journey: Never Ending Story

DevOps is not an “end game”, it’s a journey with continuous improvement

Need to monitor and revise all your processes, methodologies and practices

Automation is key because grant time to your team to be able to improve

Embrace change and failure. Fail fast!

Continuous improvement on your solution working on a small batch strategy

Crucial practices: Communication, Sharing, Innovate

DevOps Journey: Plan

Define clearly what you (your team) will do on each iteration

Your plan must focus on your customer. How do you can bring value?

Be prepared to revise your initial plan at the beginning of each iteration

Tracking is crucial, you need to understand where you are

- Doesn't mean micro-management
- Autonomy and Accounting!

Strong connection with agile methodologies

- Execute in iterations
- Embrace change
- But, not mandatory! You can do DevOps with traditional methodologies

DevOps Journey: Build

You have a great plan but without execution is only a great plan…

Create your code collaboratively using a well define and well-known process

Source control is key. Versioning, auditing, single source of truth

Use collaboration

- Peer reviewing (Pull requests)
- Branch strategy aligned with your team maturity and skills

What is code?

- Source code
- Infra as Code
- Pipelines as Code
- Everything as Code!

DevOps Journey: Continuous Integration (CI)

Create your package with one goal in mind: Production!

Automation is key

Improvement of build process is one crucial step to be agile

Tasks to be executed

- Compilers, Transpillers, …
- Create deployable package: Binaries, Zip files, container images
- Testing

Testing types

- Unit testing
- Integration testing
- Functional testing
- Performance (load) testing

DevOps Journey: Continuous Integration (CI)

Bring quality to your code

SAST (Static Application Security Testing)
Code static analysis (Performance, maintainability, …)
Credential Scanning
Software Composition Analysis

DevOps Journey: Deploy

Promote between environments until reach the only place where you bring value to your end users: Production!

Automation is key!

- Deployments tends to be repetitive tasks
- Human intervention makes your process more error prone
- Automation gives you consistency and reliability (and performance…)
- Work with quality gates (automatic validation) and approvals (manual validation)
- Enable you to have a continuous delivery

Bring innovation with testing into production

- Canary deployments
- Blue/Green deployments
- Ring deployments

DevOps Journey: Operate

Clear view about all components of your solution on a technical view

- Infrastructure, networking, workloads, everything

Applications today tends to be fully distributed

- Application Performance Management (APM) allows to have full understanding
- Distributed tracing allow to know where your requests flow

Observability

- Knows how your system is performing
- Well defined processes to operate the system
- Mix of manual and automatic intervention

Focus your Operation team on this phase to bring optimization

- Better performance
- Better reliability
- Better security
- Less costs

DevOps Journey: Continuous Feedback

Learn about your solution on a user view to understand what is “value”

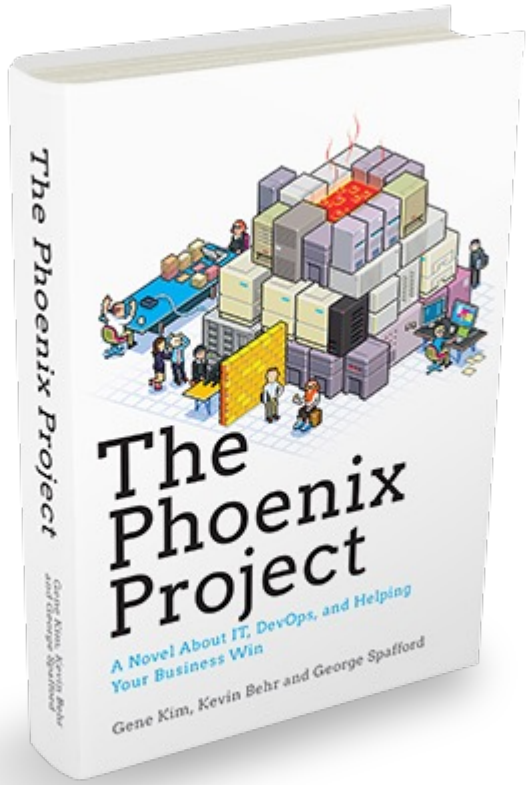
- How your users use your solution
- Collect usage metrics
- When possible, ask for feedback

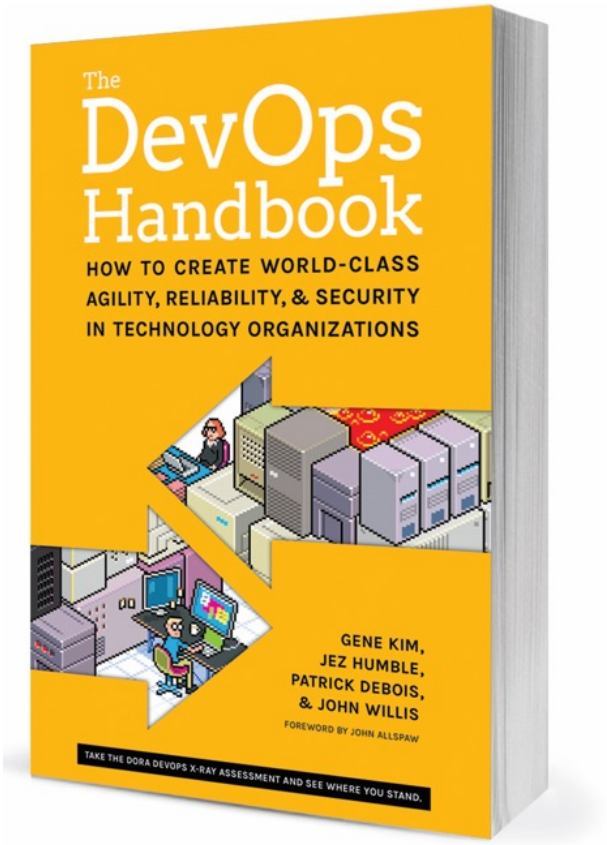
Sometimes your perception of “value” is completely the opposite from users view

Embrace change!

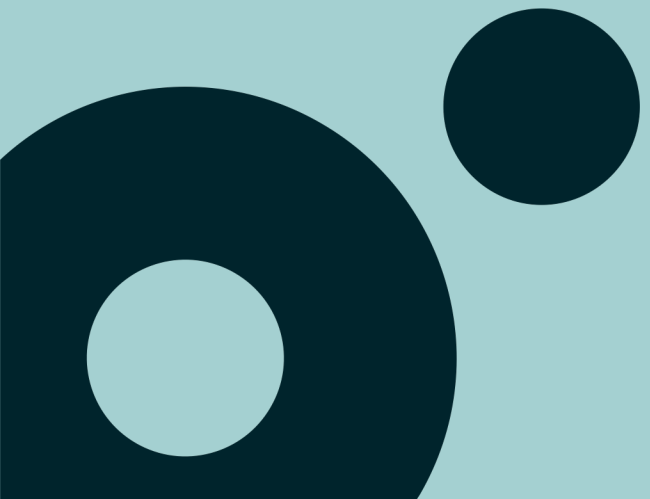
Feed your next iteration with all your learning from this phase

DevOps Journey: Books Recommendation





DevOps Challenges



DevOps Main Goals

Shorter development cycles

Keep you competitive

Produce value, quality and secure deliveries keeping your pace

Collaboration between all teams (Dev, Ops, Security, Infra, Marketing, Business,…)

Remove chaos development

Implement practices (opposite of heavy processes) with proven records

Better understanding of your solutions/products

DevOps Challenges

DevOps Culture

Practices

Metric

Tooling

moOngy.

DevOps Challenges

DevOps Culture

Practices

Metrics

Tooling

moOngy.

DevOps Challenges: Culture

Cultural changes on your organization is a key factor for DevOps success

People are the most important part of this journey

Only possible when you have your people engaged

Culture you don't "teach & learn", you live it and spread to other people

Take time, patience and effort to have it in place

Needs support from everyone

Autonomy, openness, accountability and improvement!

You may start small (on a team, project) and make it bigger

Everything start with changing habits

DevOps Challenges: Habits

Be customer obsessed

Iterate over pain

Production first mindset

Team Autonomy + Organization Alignment

Shift-left quality

Infrastructure as Flexible Resources (remove waste)

Learn how to fail fast

DevOps Challenges

DevOps Culture

Practices

Metrics

Tooling

moOngy.

DevOps Challenges: Practices

Agile Methodologies

Perfect match with infinite loop
Embrace changes
Continuous improvement already in place

Version Control System

Branch strategies
Peer reviewing
Automatic validation
Single source of truth

Continuous Integration (CI) and Continuous Delivery (CD)

Automations to bring consistency, performance and reliability
Automate quality gates and validation
Smaller probability of failures
Live documentation of your processes

DevOps Challenges: Practices

Shift-Left

Start to care of important topics as soon as possible
Testing
Security
Quality
Integration

Shift-Right

Use production environments at your benefit
Test in production! Ring deployments, canary deployments
Focus on reliability and high availability (chaos engineering, penetration testing)

Everything as code

Infra as Code, Pipeline as Code, Docs as Code
Better way to clearly define what is a version of your product

DevOps Challenges

DevOps Culture

Practices

Metrics

Tooling

moOngy.

DevOps Challenges: Metrics

Remember, goal is to continuous deliver value to end user

Traditional metrics: Lines of code (LoC), hours consumed, …

How you measure if you are delivering value looking into LoC?

Something one LoC makes much more impact than thousand of LoC

Measure outcome not activities!

New metrics to check teams productivity

- Deployment Frequency
- Mean Time to Recover
- Lead Time to Change
- Change Failure Rate

DevOps Challenges: Metrics

Deployment Frequency

How many times we bring more value to the end user?

Proxy metric for batch size

The more frequently you deploy the smaller the size of the batch

Small batch sizes reduce cycle times, reduce risk and overhead, improve efficiency, increase motivation and urgency and reduce costs

Lead Time to Change

How much time between start coding and be available in production?

Lead time is the time it takes to go from a customer making a request to the request being satisfied

Shorter lead times enable faster feedback

DevOps Challenges: Metrics

Mean time to Restore (MTTR)

How much time to recover from an error in production?

Reliability is traditionally measured as time between failures but in a modern software organization is inevitable

Reliability is now measured by how long it takes to restore service when a failure occurs

Change Failure Rate

Which percentage of failures in production that needs urgent recovery?

This metric looks at the percentage of changes made to production that fail

DevOps Challenges: Metrics

	Elite	High	Medium	Low
Deployment frequency	**On-demand** (multiple deploys per day)	Between **once per day** and **once per week**	Between **once per week** and **once per month**	Between **once per month** and **once every six months**
Lead time for changes	Less than **one day**	Between **one day** and **one week**	Between **one week** and **one month**	Between **one month** and **six months**
Time to restore service	Less than **one hour**	Less than **one day**	Less than **one day**	Between **one week** and **one month**
Change failure rate	0-15%	0-15%	0-15%	46-60%

DevOps Challenges

DevOps Culture

Practices

Metrics

Tooling

moOngy.

DevOps Challenges: Tooling

To achieve DevOps goal with need tools to support you

Supporting tools means they support your people and your processes

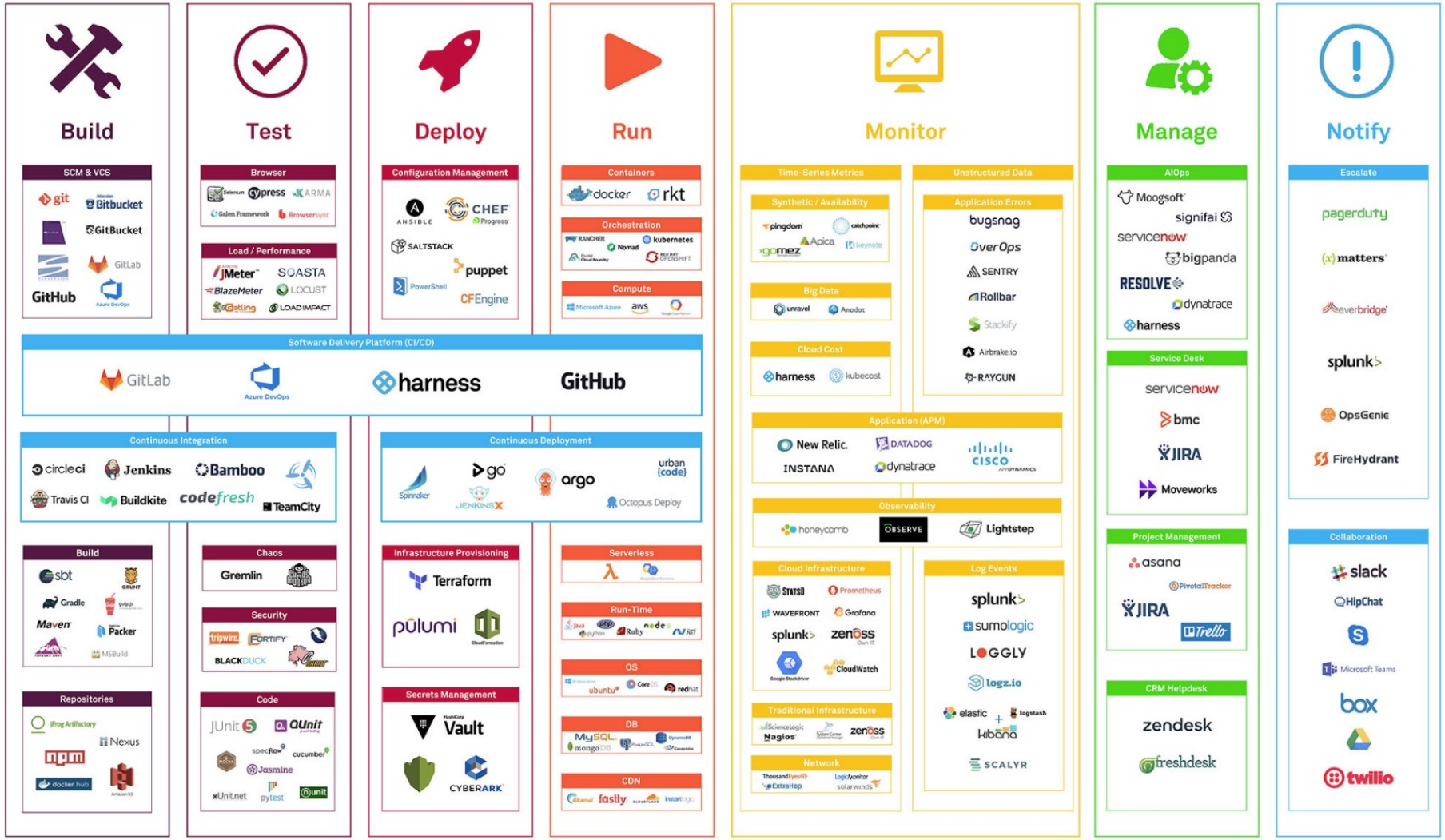
You need to select which tool better fit for you

- Understand your team skills and maturity
- Find a fit with your process
- Only in specific scenarios you may favor tools on top of people & process

Benefits from using specialized tools

- Agility
- Automation
- Auditing
- Security and permission

DevOps Challenges: Tooling Landscape



moOngy.

DevOps Challenges: Tooling Landscape

Tooling landscape is HUGE!

Every week “the next new thing” appears

It’s really important to be always aware of new things

But, it’s more important to feel comfortable with your tooling choice

Most of this tooling are really open for integration

And remember, fail fast! If some tool don’t deliver, try another one.

DevOps Challenges: Training Tooling Landscape

DevOps Platform	GitHub
Planning	GitHub Issues / GitHub Discussions
Source Control	GitHub Repos
CI/CD	GitHub Actions
Secure DevOps	GitHub Advanced Security
Infra as Code	Terraform
Testing	Playwright
Infra Provider	Azure

moOngy.
Minds on the move

Rua Sousa Martins, nº 10

1050-218 Lisboa | Portugal