DevOps – The Definitive Guide

Contents

[Library Overviews 2](#_Toc66082577)

[PagoogCommon 2](#_Toc66082578)

[PagoogCore 2](#_Toc66082579)

[PagoogDebug 2](#_Toc66082580)

[PagoogLog 2](#_Toc66082581)

[PagoogRender 2](#_Toc66082582)

[Creating and Running an Application 3](#_Toc66082583)

[Where’s my **main** function? 3](#_Toc66082584)

# The Essence of DevOps

Edited: 20/03/2021

## What is DevOps?

DevOps is the combination of principles, practices and tools designed to improve an organisation’s ability to deliver quality software and services faster than traditional processes. For software developers, think of software development as a link in a chain and DevOps as the thing that makes the links stronger chain shorter and chain thicker.

## Why do I need DevOps?

You don’t. But if you want to improve your “ability to deliver quality software and services faster” then DevOps is your friend.

## How is DevOps implemented?

DevOps is implemented incrementally by introducing principles, practices and tools to the overall workflow. These will change depending on your goals and the context you’re working in, but will all feed back into the idea of DevOps as a concept.

# The CALMS Framework

Edited: 20/03/2021

Created by Jez Humble, the CALMS framework is a way for an organisation to assess how DevOpsy they are using 5 tenants.

### Culture

The idea of DevOps needs to be firmly supported and desired by the organisation if there is to be any chance at success.

### Automation

Anything that can be automated should be automated – within reason.

### Lean

Eliminate waste and streamline processes to make things clearer and easier.

### Measurement

The organisation should be devoted to collecting and analysing data on their own capabilities to see where strengths and weaknesses lie.

### Sharing

A culture of openness and sharing between teams and individuals should be highly encouraged.

# Team Topologies

Edited: 20/03/2021

## Stream-aligned teams

Stream-aligned teams focus on a single stream of work. A product, feature, user journey or persona. They are empowered to build and deliver customer value as quickly and safely as possible without relying on handing parts of the work off to other teams.

## Platform teams

Platform teams are designed to work with autonomy too, however, the main focus is to deliver internal tools and services that stream-aligned teams (and others) can make use of. This is usually what provides cohesive experiences across products made by the same organisation.

## Subsystem teams

Subsystem teams are teams dedicated to working on specialised areas that might require a specific set of skills or knowledge. The idea here is to remove the burden on stream-aligned teams that may come from including work that is outside of their day-to-day scope.

## Enabling team

Enabling teams are typically responsible for research and experimentation into tooling, frameworks or ecosystems that can make the lives of other teams easier. This team takes this weight off the stream-aligned team so that they can focus on meeting critical deadlines whilst also making use tools, frameworks and ecosystems they otherwise wouldn’t be.

# Metrics

Edited: 20/03/2021

There are 4 metrics that are very beneficial to measure DevOps performance against. You can always add more for your own organisation but these 4 are fundamental.

### Lead time for changes

This is how long it takes for a code change to go from checked-in to releasable.

Typically, high performing teams will measure this in hours, with days and weeks indicating lower performance.

Developers should be able to receive fast feedback on the quality of the code they have written to remediate defects sooner.

Long lead times of days indicate a reliance on manual testing over automated.

### Change failure rate

The percentage of code changes that require hot fixes or remediation after release.

High performing teams will have a rate of 0-15%.

The same things used to reduce lead times often also decrease change failure rate.

### Deployment frequency

How often the new code released to the customers.

High performing teams will be able to release on demand, with no interruption to workflow.

Minimising human intervention in this process is what takes lower performing teams into the higher percentiles.

### Mean time to recovery

How long it takes to recover from a partial service interruption or total failure.

Higher performing teams may only take an hour to get back up and running.

This all depends on how quick failures can be identified, how easy it is to rollback changes and how quick it is to fix and redeploy changes.

This is preferred to mean time between failures as instead of waiting until a release is perfect teams continuously deploy releases which encourages blameless retrospectives to grow and learn from.