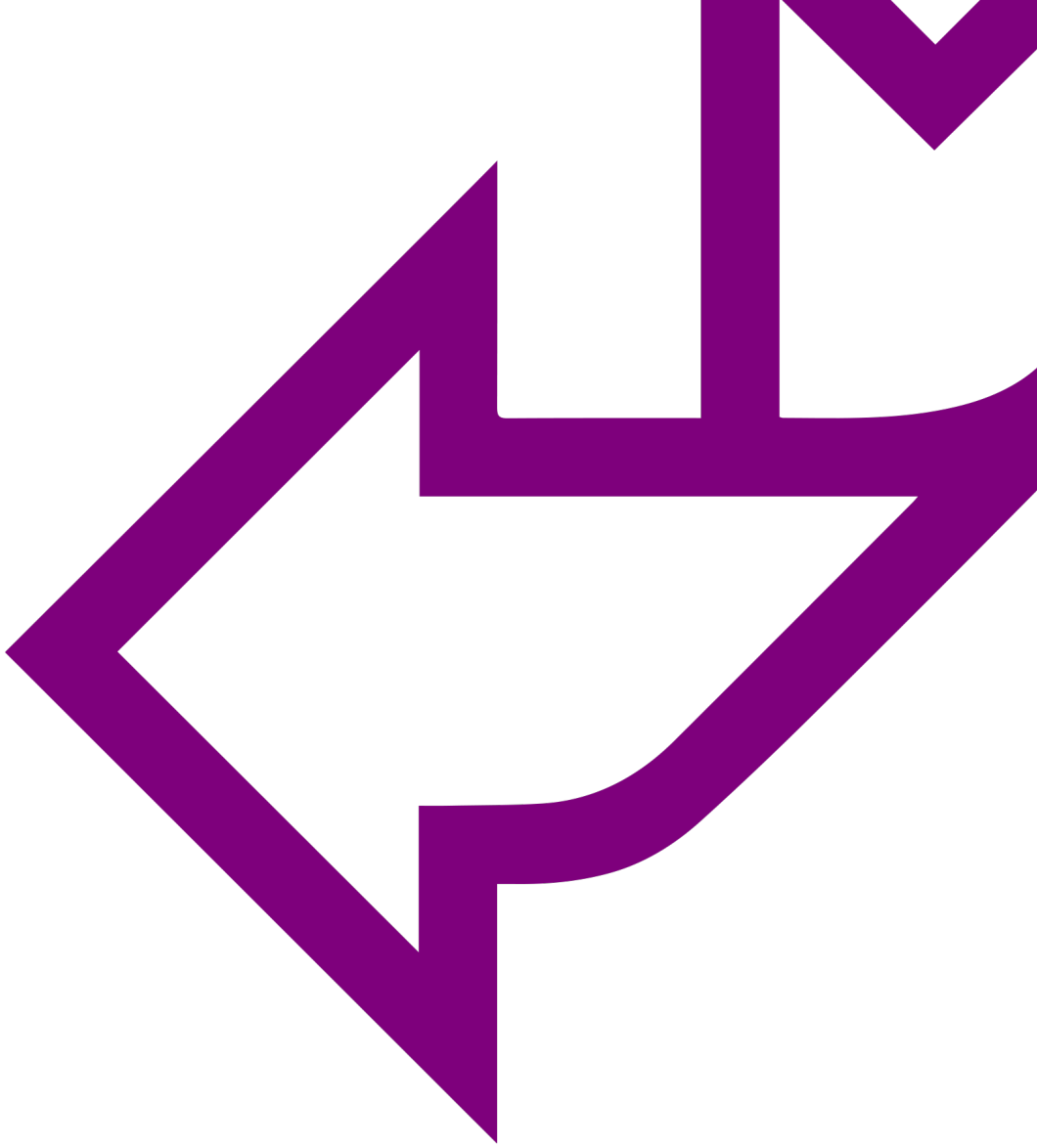
POWERING POTENTIAL

Introducing DevOps and Continuous Integration



Module 5 – Continuous Integration

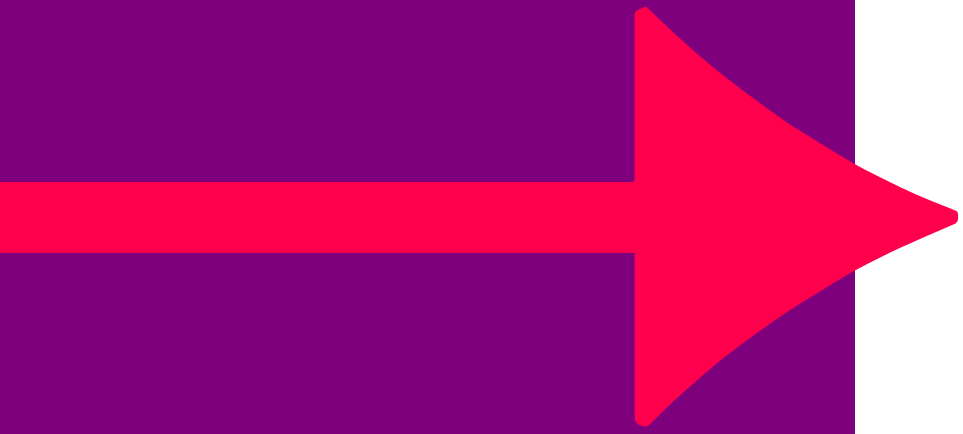
What is DevOps?

The term **DevOps** can be a difficult one to define as different organisations have varying approaches to implementing DevOps.

It is best described as a cultural approach to software development project structure with a particular philosophy designed to achieve the following:

- Increased collaboration
- Reduction in silos
- Shared responsibility
- Autonomous teams
- Increase in quality
- Valuing feedback
- Increase in automation

IT BEFORE DEVOPS

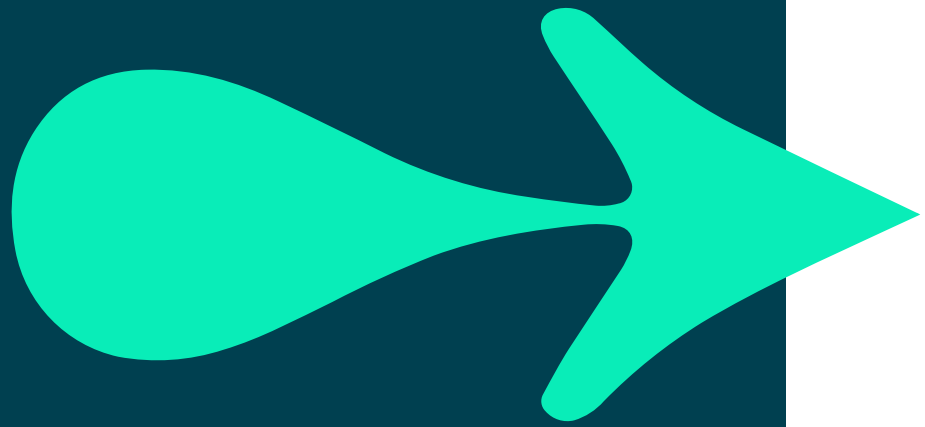


Traditionally, software companies are structured as separate, stratified teams for development, quality assurance (sometimes known as testers), security, and operations.

These teams tend to have varying (and sometimes conflicting!) goals which, when paired with poor communication between them, can regularly result in work that is out of sync with other parts of the organisation. These isolated teams are referred to as silos.

As a result, this structure regularly results in slower releases, wasted time (and therefore money), and blame cultures, where production problems become the fault of another team.

DEVOPS AND AUTOMATION: INTEGRATION, DELIVERY AND DEPLOYMENT



Automation around process is a big part of DevOps, and this typically focuses around three main areas:

- **Continuous Integration** - Integration and testing of new features into software
- **Continuous Delivery** – Deploying safe-working code into a staging environment
- **Continuous Deployment** - Deployment of applications to a customer for usage

Continuous Integration, Delivery, Deployment

Continuous Integration

Continuous Integration (CI) is a software dev practice that revolves around merging changes often

Different features are combined, or integrated, frequently to expose issues quickly

Automating the integration of code, the building of software, and testing of software is key

Continuous Delivery

Continuous delivery extends on CI, ensuring that you can release new changes to customers quickly on a schedule

We automate the deployment to a staging environment – an environment that mimics the production environment as closely as possible

Once we've verified that it works in this staging environment, it should work in a production environment

Continuous Deployment

Continuous Deployment extends even further on Continuous Delivery, automating deployment into a customer-facing environment

As you can probably assume, this is quite the difficult and stressful task!

If anything is wrong with a feature, it could be seen by customers

Continuous Integration, Delivery, Deployment

Continuous Integration

Continuous Integration (CI) is a software dev practice that revolves around merging changes often

Different features are combined, or integrated, frequently to expose issues quickly

Automating the integration of code, the building of software, and testing of software is key

Continuous Delivery

Continuous delivery extends on CI, ensuring that you can release new changes to customers quickly on a schedule

We automate the deployment to a staging environment – an environment that mimics the production environment as closely as possible

Once we've verified that it works in this staging environment, it should work in a production environment

Continuous Deployment

Continuous Deployment extends even further on Continuous Delivery, automating deployment into a customer-facing environment

As you can probably assume, this is quite the difficult and stressful task!

If anything is wrong with a feature, it could be seen by customers

Continuous Integration, Delivery, Deployment

Continuous Integration

Continuous Integration (CI) is a software dev practice that revolves around merging changes often

Different features are combined, or integrated, frequently to expose issues quickly

Automating the integration of code, the building of software, and testing of software is key

Continuous Delivery

Continuous delivery extends on CI, ensuring that you can release new changes to customers quickly on a schedule

We automate the deployment to a staging environment – an environment that mimics the production environment as closely as possible

Once we've verified that it works in this staging environment, it should work in a production environment

Continuous Deployment

Continuous Deployment extends even further on Continuous Delivery, automating deployment into a customer-facing environment

As you can probably assume, this is quite the difficult and stressful task!

If anything is wrong with a feature, it could be seen by customers

EXERCISE 1 – CI RESEARCH

In groups, research online about a company that utilises DevOps, CI and CD principles within their business

- Note down who the business are, what their main trade/service is, etc.

Answer the following questions where possible:

- Why did the business adopt CI/CD?
- How has it improved their development processes?
- What tools do they use that you recognise, if any?

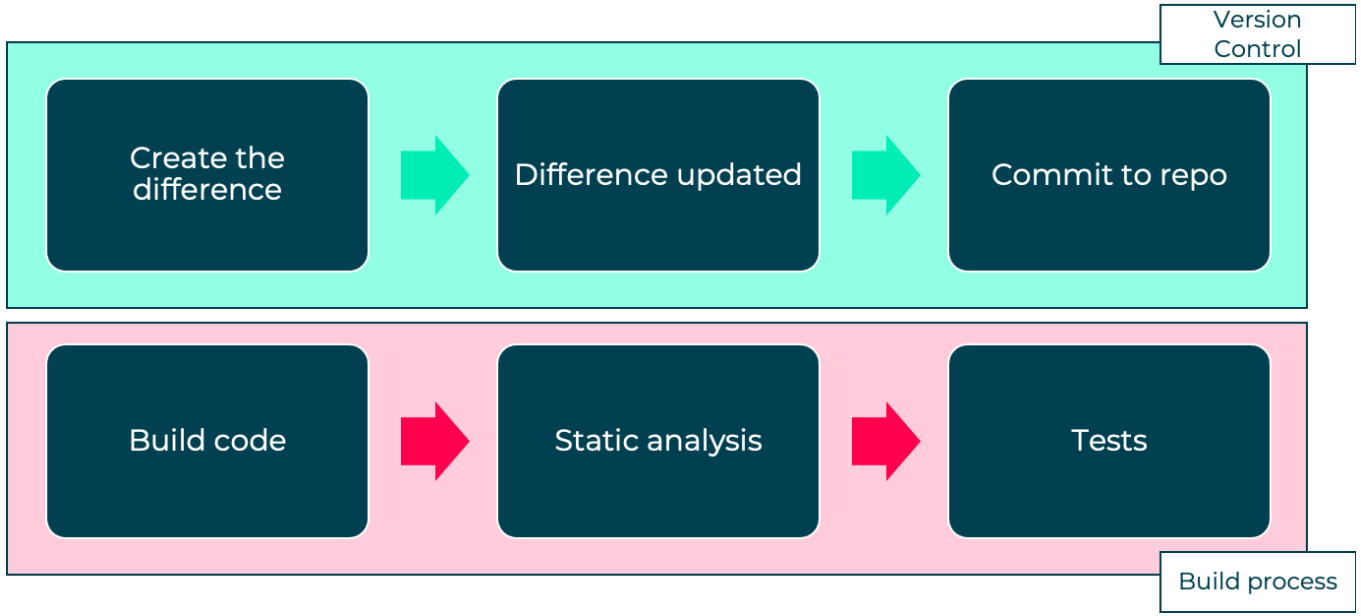
Share your research and thoughts with the rest of the room

CI IN ACTION – FACEBOOK

Facebook is one of the biggest social media platforms and has billions of monthly active mobile users

Christian Legnitto discussed the way Facebook developed their mobile platform in 2014

- Number of different teams – messages, photos, Android, iOS, events
- Have over 100,000 commits to their Git repository
- 300 contributes with over 150,000 files




END OF SECTION