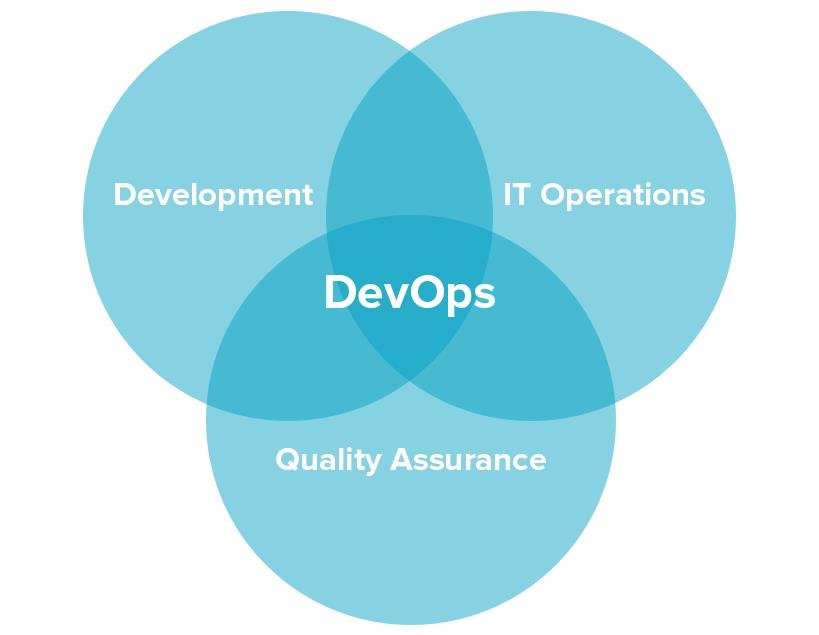
**EXP-1**

Aim: To understand DevOps: Principles, practices and DevOps Engineer role and responsibilities

## What Is DevOps?

DevOps stands for development and operations. It’s a practice that aims at merging development, quality assurance, and operations (deployment and integration) into a single, continuous set of processes. This methodology is a natural extension for Agile and continuous delivery approaches.



What DevOps looks like.

But DevOps isn’t merely a set of actions. It’s more of a culture or even a philosophy that fosters cross-functional team communication. One of the main benefits of DevOps is that it doesn’t require substantial technical changes being rather oriented to changing the way a team works. Teamwork is a crucial part of DevOps culture: The whole success of a process depends on it, and there are principles and practices that DevOps teams use.

### **DevOps Principles**

In short, the main principles of DevOps are automation, continuous delivery, and fast reaction to feedback. You can find a more detailed explanation of DevOps pillars in the **CAMS**acronym:

**C**ulture represented by human communication, technical processes, and tools

**A**utomation of processes

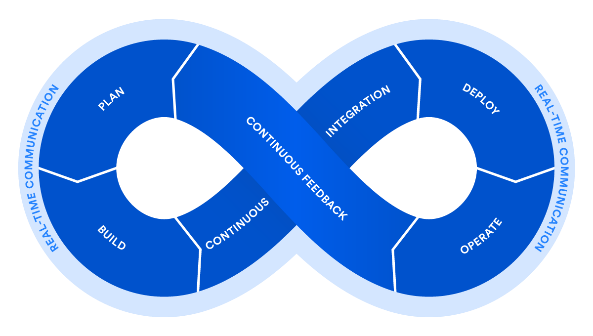
**M**easurement of KPIs

**S**haring feedback, best practices, and knowledge

Adherence to these principles is achieved through a number of DevOps practices that include continuous delivery, frequent deployments, QA automation, validating ideas as early as possible, and in-team collaboration.

## DevOps Model and Practices

DevOps requires a delivery cycle that comprises planning, development, testing, deployment, release, and monitoring with active cooperation between different members of a team.

A DevOps lifecycle. Source: [*Atlassian*](https://www.atlassian.com/devops).

To break down the process even more, let’s have a look at the core practices that constitute the DevOps:

### Agile Planning

In contrast to traditional approaches of project management, Agile planning organizes work in short iterations (e.g. sprints) to increase the number of releases. This means that the team has only high-level objectives outlined while making detailed planning for two iterations in advance. This allows for flexibility and pivots once the ideas are tested on an early product increment. Check [our Agile infographics](https://www.altexsoft.com/infographics/agile-project-management-methods/?utm_source=DZone&utm_medium=referral) to learn more about the different methods applied.

### Continuous Delivery and Automation

[Continuous delivery](https://www.altexsoft.com/blog/business/continuous-delivery-and-integration-rapid-updates-by-automating-quality-assurance/?utm_source=DZone&utm_medium=referral), detailed in our dedicated article, is an approach that merges development, testing, and deployment operations into a streamlined process as it heavily relies on automation.

**Development.**Engineers commit code in small chunks multiple times a day for it to be easily tested.

**Continuous automated testing and integration.**A quality assurance team sets committed code testing using automation tools like Selenium, Ranorex, UFT, etc. If bugs and vulnerabilities are revealed, they are sent back to the engineering team. This stage also entails version control to detect integration problems in advance.

**Continuous deployment.**At this stage, the code is deployed to run in production on a public server. Code must be deployed in a way that doesn’t affect already functioning features and can be available for a large number of users.. The most popular are Chef, Puppet, Azure Resource Manager, and Google Cloud Deployment Manager.

**Continuous monitoring.**The final stage of the DevOps lifecycle is oriented to the assessment of the whole cycle. The goal of monitoring is detecting the problematic areas of a process and analyzing the feedback from the team and users to report existing inaccuracies and improve the product’s functioning.

### Infrastructure as Code

Infrastructure as a code (IaC) is an infrastructure management approach that makes continuous delivery and DevOps possible. It entails using scripts to automatically set the deployment environment (networks, virtual machines, etc.) to the needed configuration regardless of its initial state.

### Containerization

The next evolutionary stage of virtual machines is containerization. Virtual machines emulate hardware behavior to share computing resources of a physical machine, which enables running multiple application environments or operating systems (Linux and Windows Server) on a single physical server or distributing an application across multiple physical machines.

### Microservices

The microservice architectural approach entails building one application as a set of independent services that communicate with each other, but are configured individually. Building an application this way, you can isolate any arising problems ensuring that a failure in one service doesn’t break the rest of the application functions.

### Cloud Infrastructure

Today most organizations use [hybrid clouds](https://www.altexsoft.com/blog/cloud/preparing-for-hybrid-cloud/?utm_source=DZone&utm_medium=referral), a combination of public and private ones. But the shift towards fully public clouds (i.e. managed by an external provider such as AWS or Microsoft Azure) continues. While cloud infrastructure isn’t a must for DevOps adoption, it provides flexibility, toolsets, and scalability to applications.

DevOps Engineer Responsibilities

|  |  |
| --- | --- |
| **Delivery management** | The process of delivery management and the DevOps lifecycle has a lot of points of intersection since both of these processes take place at each of the product development stages. DevOps engineers make sure that tech team members are clear on goals, tasks, and deadlines. |
| **IT infrastructure design and improvement** | DevOps specialists are always reviewing the IT infrastructure design and improvements with the aim of creating more effective collaboration and reducing risks. |
| **Performance testing** | Since continuous delivery is one of the DevOps pillars, there is a need to test each piece of software created before its integration into the existing business processes. |
| **Automation** | A lot of DevOps engineer responsibilities imply repeated tasks with each of the software pieces. What do DevOps engineers do in this case? They automate them with the help of [DevOps automation tools](https://www.cprime.com/cloud/devops-solutions/devops-tooling/) or [create custom automation scripts](https://www.cprime.com/software-development-and-support/custom-software-development/). |
| **Release cycle optimization** | The continuous delivery approach requires DevOps engineers to constantly search for ways to optimize and improve release cycles, eliminate time drains and deliver better products faster. |
| **Monitoring and reporting** | Since continuous feedback is one of the DevOps principles, this specialist needs to control the process, report on the mistakes, and find ways to avoid them in a new cycle. |
| **Security control** | Most DevOps processes take place in the cloud, so there is a need for security control, which is one more task of a DevOps engineer. Despite being the last job on the list, it is one of their most important responsibilities. |