UTT
UNIVERSIDAD TECNOLÓGICA DE TIJUANA
GOBIERNO DE BAJA CALIFORNIA

**TOPIC:**

Introduction to DevOps

**STUDENT:**

Maciel Leyva Ariana Lizeth

**GROUP:**

10A

**SUBJECT:**

Comprehensive Mobile Development

**TEACHER:**

Ray Brunett Parra Galaviz

**Tijuana, Baja California, January 8th of 2025**

# “Introduction to DevOps”

DevOps is the combination of practices and tools designed to increase an organization’s ability to deliver applications and services faster than traditional software development processes. This speed enables organizations to better serve their customers and compete more effectively in the market.

In simple terms, DevOps is about removing the barriers between traditionally siloed teams, development and operations. Under a DevOps model, these teams work together across the entire software application life cycle, from development and test through deployment and operations.

## How does DevOps work?

A DevOps team includes developers and IT operations working collaboratively throughout the product lifecycle, in order to increase the speed and quality of software deployment. It’s a new way of working, a cultural shift, that has significant implications for teams and the organizations they work for. Under a DevOps model, development and operations teams are no longer “siloed.” Sometimes, these two teams merge into a single team where the engineers work across the entire application lifecycle from development and test to deployment and operations and have a range of multidisciplinary skills.

DevOps teams use tools to automate and accelerate processes, which helps to increase reliability. A DevOps toolchain helps teams tackle important DevOps fundamentals including continuous integration, continuous delivery, automation, and collaboration. DevOps values are sometimes applied to teams other than development. When security teams adopt a DevOps approach, security is an active and integrated part of the development process. This is called DevSecOps.

## What is a DevOps methodology?

DevOps methodologies are a collection of procedures and strategies that enable businesses to produce higher-quality, reliable software quickly. These approaches

are made to encourage communication between the development and operations teams, automate routine tasks, and support continuous delivery and improvement.

The DevOps methodology may differ between organizations, teams, and purposes but typically include:

**Continuous Integration (CI)**

This is a technique where developers frequently merge their code changes into a main repository. Automated builds and tests are executed following the addition of modifications. The objective is to decrease the time needed to validate and deploy new software updates, enhance software quality, and promptly identify and fix defects.

**Continuous Delivery (CD)**

This is a perpetuation of continuous integration. After the build step, it then automatically deploys all code alterations to a testing and/or production environment. By using this procedure, you can have a deployable build artifact that is prepared to go live at any time, enabling a steady stream of improvements for consumers.

**Monitor and logging**

This entails keeping track of and logging system activity in order to assess the health of the system, identify issues, and resolve them. Effective user behavior, performance of the system, and potential error insights can all be gained by effective monitoring and logging.

**Automation**

Automation applies to many phases of the software delivery lifecycle, including infrastructure setup, integration, testing, and deployment. Automation speeds up the software delivery process, minimizes the possibility of human error, and helps remove manual, repetitive operations.

**Microservices**

This is a type of architectural design in which a single application is divided into a number of smaller services, each of which operates independently and interacts with the others via a well-defined interface. Teams can deploy, scale, upgrade, and debug services independently using this method, which can speed up these procedures and lower the possibility of broad system failures.

**Infrastructure as Code (IaC)**

IaC in DevOps is the practice where infrastructure is organized and controlled by code and software development methods such as version control and continuous integration. This enables the management of infrastructure by developers and operations teams in manners comparable to that of application code. IaC can greatly increase the process of setting up and updating infrastructure consistently and repeatably.

**Collaboration and sharing**

DevOps promotes a collaborative environment in which developers and operations professionals cooperate throughout the application lifecycle. Sharing responsibility, criticism, and understanding can result in greater comprehension, quicker problem-solving, and increased productivity. DevOps methodologies are not siloed but rather are interwoven practices and strategies. Applying DevOps to an organization's needs is more than just adhering to these standards.

Implementing DevOps methodologies necessitates a cultural shift toward cooperation, transparency, and shared responsibility which could also call for more adjustments to roles, expectations, competencies, resources, and business structure.

## What is the DevOps lifecycle?

The DevOps lifecycle is a continuous workflow that involves the constant interaction of a variety of stages. These stages are designed to create better collaboration between development, operations, and quality assurance teams in order to accelerate the delivery of software products, ensure their quality, and enhance customer satisfaction.

## The DevOps lifecycle includes:

### Planning

At this stage; teams define the system requirements, create a project plan, and recognize the needs and expectations. The purpose of this stage is to guarantee that all stakeholders have a firm grasp of the requirements and objectives of the project.

### Coding

At this point, programmers begin writing the application's code. The standards and needs established during the planning stage serve as the basis for the coding. Version control systems are used by developers to maintain track of multiple code versions as well as a variety of coding tools.

### Building

When the code is finished, it is constructed or compiled into an executable form that can be used to run on a computer. This process is automated using automated build tools, which can improve effectiveness and lower error rates.

### Testing

At this stage, the software will be tested after it has been developed to identify any faults or problems and solve them. This analysis covers functional testing, integration

testing, and other kinds of testing. In order to boost efficiency and automate repetitive testing activities, automation technologies are frequently utilized in this stage.

**Releasing**

The software is prepared for release after testing. This entails putting the program into production and making end users aware of it. In order to automate the release process and enable regular and dependable software releases, continuous delivery and continuous deployment procedures are frequently utilized.

**Deploying**

This stage involves the installation of the released software into the production environment. The purpose is to make sure the software functions properly in a real-world setting. This process is automated by continuous deployment, lowering the probability of deployment failures and facilitating quicker and more reliable software delivery.

**Operating**

To ensure the software keeps working appropriately after deployment, it needs to be maintained and monitored. This includes keeping track of the software's operating performance, finding any bugs or issues, and correcting any problems that come up.

**Monitoring**

The last stage includes regular monitoring of the software being used in production with the objective to find and addressing any problems before they have a negative impact on users. This stage may also compile data that may be used to enhance the software versions in the future. Also, system performance, user behavior, error rates, and other indicators can all be monitored.

The DevOps lifecycle is a continual loop that is reflected in the non-linear design of the process. Continuous improvements and optimizations result from the monitoring

stage and feed back into the planning stage for the following iteration. Communication and cooperation are essential at every level of the DevOps lifecycle. The purpose is to dismantle team silos and promote improved collaboration and understanding.

**What are the benefits of DevOps?**

Several key practices can help organizations innovate faster by automating and streamlining the software development management process.

**Development velocity:** DevOps automation and practices let you innovate faster, adapt to changing markets better, and increase the pace of releases, so you can improve your product faster and build competitive advantage.

**Reliability:** DevOps practices like continuous integration and continuous delivery can ensure the quality of application updates and infrastructure changes, so you can reliably deliver at a more rapid pace while maintaining an optimum experience for end users.

**Improved collaboration:** Communication and collaboration are keystones of DevOps practices. Automation of the software delivery process establishes collaboration by physically bringing together the workflows and responsibilities of development and operations. Communication across developers, operations, and even other teams, such as marketing and sales, allows all parts of the organization to align more closely on goals and projects.

**Security:** You can adopt a DevOps model without sacrificing security by using automated, integrated security testing tools.

**Sources**

GitLab. (2023, January 25th). *What is DevOps??* | *GitLab*. GitLab. Retrieved January 8th, 2025, of https://about.gitlab.com/topics/devops/

*What Is DevOps and How Does It Work?* | *Black Duck*. (s. f.). Retrieved January 8th, 2025, of https://www.blackduck.com/glossary/what-is-devops.html

Atlassian. (s. f.). *What is DevOps?* | *Atlassian*. Retrieved January 8th, 2025, of https://www.atlassian.com/devops

SonarSource. (2024, April 11th). *DevOps developer's guide*. Sonar. Retrieved January 8th, 2025, of https://www.sonarsource.com/learn/devops/