**DevOps Model**

**What is DevOps**

The term DevOps is the short form of **Development** and **Operations**.

It focuses on collaboration between developers and other roles.

DevOps is a practice that allows a single team to manage the entire application development life cycle, that is, development, testing, deployment, operations.

DevOps aims to shorten the system’s development life cycle while delivering features, fixes, and updates frequently in close alignment with business objectives.

DevOps is an evolution of the **Agile(Atik,çevik) Model** of software development.

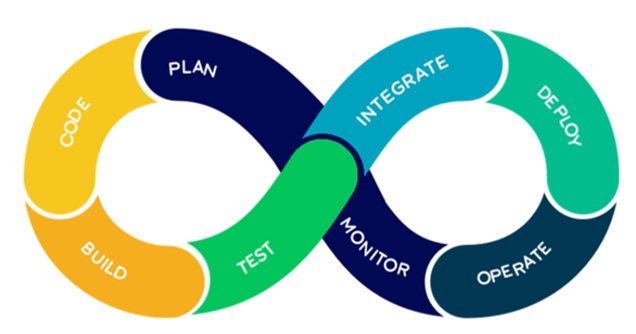
As the Agile model addressed the gap between clients and developers, DevOps addressed the gap between Developers and Operations.

The development team will submit the application to the operations team for implementation.

The operations team will monitor the application and provide relevant feedback to developers.

**DevOps Phases**

According to DevOps practices, the workflow in software development and delivery is divided into **8** phases.



**Plan** - Business owners and software development team discuss project goals and create a plan.

**Code** - Programmers then design and code the application and use tools like Git to store application code.

**Build** - Build tools like Maven and Gradle, take code from different repositories and combine them to build the complete application.

**Test** - Application is tested using automation testing tools like Selenium and Junit to ensure software quality.

**Integrate** - When testing is complete, new features are integrated automatically to the already existing codebase.

**Deploy** - Application is packaged after release and deployed from the development server to the production server.

**Operate** - Once the software is deployed, the operations team performs activities such as configuring servers and provisioning them with the required resources.

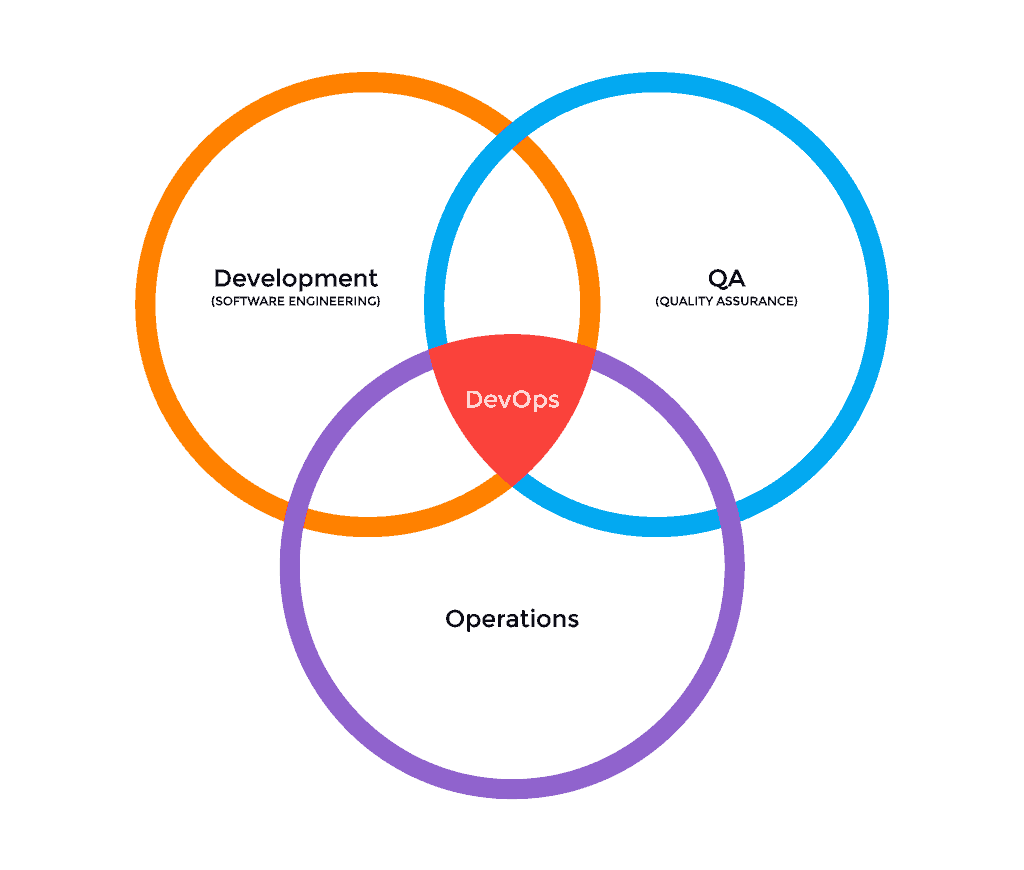
**Monitor** - Monitoring allows IT organizations to identify specific issues of specific releases and understand the impact on end-users.

**DevOps Model**

The DevOps is a new SDLC model that focuses on communication, collaboration, integration between Developers and Operations teams to accelerate innovation and the deployment of higher-quality and more reliable software products.

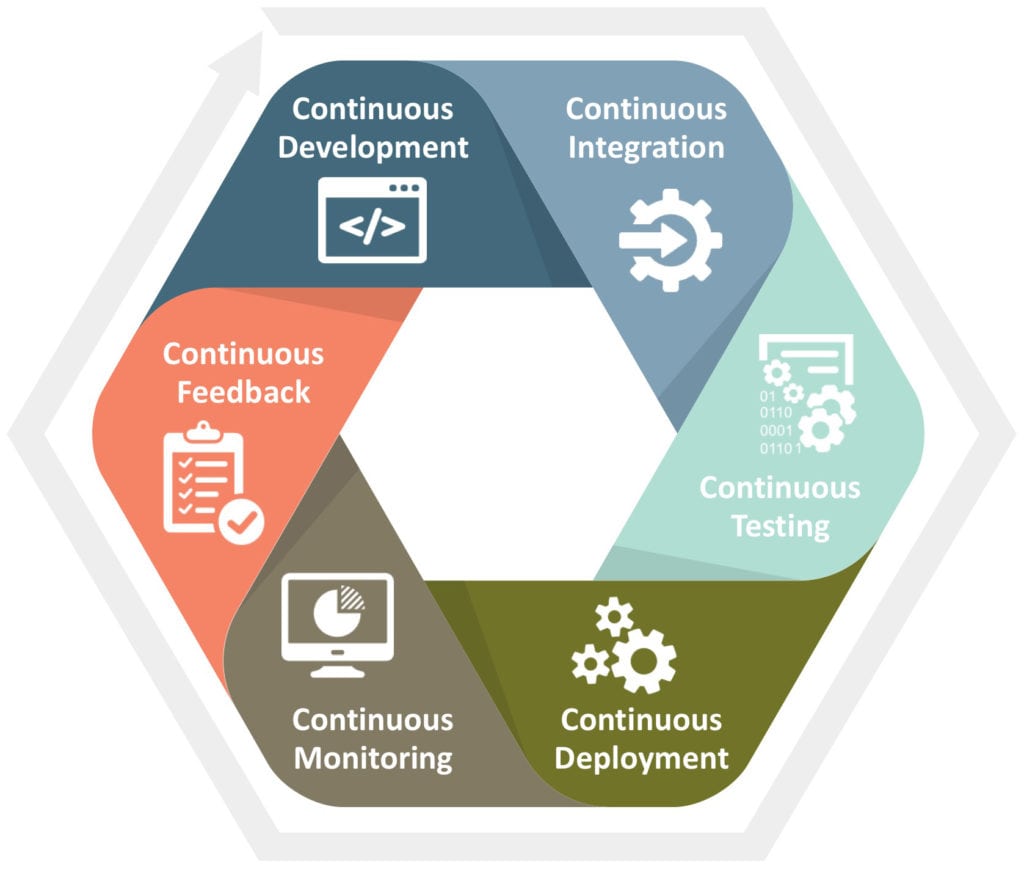
DevOps is a culture that promotes collaboration between Development and Operations teams. This allows deploying code to production faster and in an automated way. It helps to increases an organization's speed to deliver applications and services.

DevOps is the practice of operations and development engineers participating together in the entire service lifecycle, from design through the development process to production support.



**DevOps Processes**

DevOps Model consists of various stages such as continuous development, continuous integration, continuous testing, continuous deployment, continuous monitoring, and continuous feedback.



**Continuous Development**

This is the phase that involves **planning** and **coding** of the software. The vision of the project is decided during the planning phase and the developers begin developing the code for the application. There are no DevOps tools that are required for planning, but there are a number of tools for maintaining the code.

**Continuous Testing**

This is the stage where the developed software is continuously tested for bugs. For Continuous Testing, automation testing tools like Selenium, TestNG, JUnit, etc are used.

**Continuous Integration**

This stage is the **heart** of the entire DevOps life cycle. It is a software development practice in which the developers require to commit changes to the source code more frequently.

This may be on a daily or a weekly basis. Every commit is then built and this allows early detection of problems if they are present. Building code not only involves compilation but it also includes code review, unit testing, integration testing, and packaging.

The code supporting new functionality is continuously integrated with the existing code. Since there is continuous development of software, the updated code needs to be integrated continuously as well as smoothly with the systems to reflect changes to the end-users.

**Jenkins** is a very popular tool used in this phase.

**Continuous Deployment**

This is the stage where the code is deployed to the production servers.

It is also important to ensure that the code is correctly deployed on all the servers.

**Continuous Monitoring**

This is a very crucial stage of the DevOps life cycle, where it is continuously monitored the performance of the application.

Here vital information about the use of the software is recorded.

This information is processed to recognize the proper functionality of the application.

The system errors such as low memory, server not reachable, etc. are resolved in this phase.

The root cause of any issue is determined in this phase.

It maintains the security and availability of the services.

**DevOps Principles**

Some of the DevOps principles are;

* Create production-like systems for development and testing environment
* Deployments need to be iterative(yinelemeli) and frequent. Ensure a reliable and repeatable process
* Continuously monitor and validate operational quality characteristics
* Amplify(yükseltmek) feedback loop

**Advantages**

Some of the DevOps advantages are;

* Time taken to create and deliver software is reduced
* The complexity of maintaining an application is reduced
* Improved collaboration between developers and operations team
* Continuous integration and delivery ensure faster time to market

**Disadvantages**

Some of the DevOps disadvantages are;

* Technology investments in the automation tools required for DevOps are **costly** and will take a great deal of time to identify and implement.
* It needs a **specialist** who can cover each stage of the software delivery pipeline rather than investing in a smaller number of full-stack developers.
* It is easy to add new features, but more features may not always better, even if they can be implemented efficiently.