**DEVOPS CULTURE IN AN ORGANIZATION**

Most times, the development team (those whose job is to write software) in an organization are more concerned with updating their codes to maybe implement new features. In the process, something might get damaged or not work again. On the other hand, the operations staff (those in charge of deploying, monitoring and maintaining the applications) really want the application to just keep being available for the customers. They are more concerned with stability.

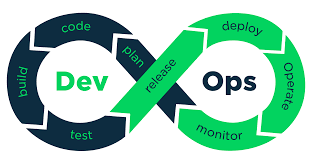
Another issue organizations used to face is when the code runs on the developers’ laptop, but not in production.

DevOps is a set of practices that helps solve these issues. The term “DevOps” was coined from two words “Development” and “Operations”.

Joseph Pellegrini defines DevOps as a set of cultural principles centred on the concepts of cross-fertilizing expertise between software development and infrastructure operations.

DevOps is a software development strategy/ methodology that helps encourage communication and collaboration between software engineering, operations and the business teams to deploy code to production environments faster in a repeatable and automated way.

DevOps is an evolution of the agile model of software development (where sprints are run and clients give reviews during each sprint).



In DevOps, the first stage is planning(continuous planning). Here the project goals are discussed, progress is measured, necessary changes in direction are made. Then the developers now design and code the application using SCMs like git to keep track and store code. Integration is done continuously. If there are any issues in the code, it is immediately sent back to the developers with what the issues are exactly. If there are no issues, deployment to production servers and constant monitoring follows.

The process really doesn’t end there. There is room for improvement, hence continuous feedback and optimization. Visualization tools are used to analyse customer journey/ behaviour and the impact on the business.

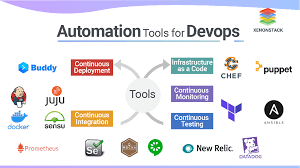
To truly achieve DevOps culture in an organization, the following principles need to be followed:

1. Encourage collaboration between development and operations to create a unilateral team that focuses on delivering common objectives. For this to happen, they need to encourage development and operations to regularly communicate, share ideas and solve problems together.
2. Impose responsibility on both groups to work as a team and be fully accountable for the application from beginning to end. One of the core principles of DevOps is the control and responsibility of services from start to finish.
3. Encourage Continuous Improvement. Organizations must continuously adapt to changing circumstances like the emergence of new technology, customer needs, or changes in legislation. In DevOps, there is a strong focus on continuous improvement to optimize performance, cost and speed of delivery.
4. Organizations must automate their processes to strive for continuous improvement with high cycle rates and the ability to immediately respond to customer feedback. Fortunately, there have been notable developments in automated tools to streamline processes like the CI/CD pipeline.
5. To fully embrace the DevOps culture, organizations must change their attitude towards failure. They should try to accept failure and learn from their mistakes.
6. Since DevOps teams are required to be involved at every stage of the software development lifecycle (from planning, building, deployment, feedback and improvement), it requires a cross-functional team where responsibility is shared. By this, practitioners with different areas of expertise all share responsibility for running your code to production.

In summary, some of the benefits of imbibing the DevOps culture in an organization are:

1. Reduced time-taken to deliver quality software.
2. Reduced complexity in maintaining an application.
3. Reduced failure rates for a new release.
4. Quicker recovery time if a new release crashes the application.
5. Improved collaboration between developers and operations team.
6. Continuous Integration and delivery to ensure faster time to the market.

**SOME OF THE IT TOOLS YOU NEED TO KNOW TO BE A GOOD DEVOPS ENGINEER**



* Git: It is a source code management tool that allows team members to collaborate remotely. It also allows one to track the progress of development work. You can also save different versions of your source code and return to a previous version when necessary.
* Gradle: It is a reliable build tool. It allows you to write code in several languages. It is also supported by several IDEs.
* Jenkins: It’s an open-source CI/CD server that allows you to automate the different stages of your delivery pipeline.
* Bamboo: Bamboo is Atlassian’s CI/CD server solution that has many similar features to Jenkins. Both are popular DevOps tools that allow you to automate your delivery pipeline, from builds to deployment. However, while Jenkins is open source, Bamboo comes with a price tag.
* Docker: It isolates applications into separate containers, so they become portable and more secure. Docker apps are also OS and platform-independent.
* Kubernetes: It’s a container orchestration platform that takes containerization to the next level. It works well with Docker or any of its alternatives. You may not need a container orchestration platform if you have just a few containers.
* Puppet: Puppet Enterprise is a cross-platform configuration management platform. It allows you to manage your infrastructure as code. As it automates infrastructure management, you can deliver software faster and more securely.
* Ansible: Ansible is a configuration management tool, similar to Puppet and Chef. You can use it to configure your infrastructure and automate deployment.
* Nagios: Nagios is one of the most popular free and open-source DevOps monitoring tools. It allows you to monitor your infrastructure so that you can find and fix problems. With Nagios, you can keep records of events, outages, and failures. You can also keep an eye on trends with the help of Nagios’ graphs and reports. This way, you can forecast outages and errors and detect security threats.

**REFERENCES**

* <https://www.youtube.com/watch?v=Me3ea4nUt0U>
* https://www.youtube.com/watch?v=a9\_oMNSgX2g
* https://opensource.com/article/20/2/devops-beginners#:~:text=In%20simple%20terms%2C%20DevOps%20is,critical%20aspect%20of%20the%20journey
* https://www.atlassian.com/team-playbook/examples/devops-culture#:~:text=DevOps%20culture%20is%20all%20about,%2C%20and%20%22the%20business%22.
* https://raygun.com/blog/best-devops-tools/
* https://www.cmswire.com/information-management/7-key-principles-for-a-successful-devops-culture/