# **Day 1 (8th October) : Getting into DevOps**

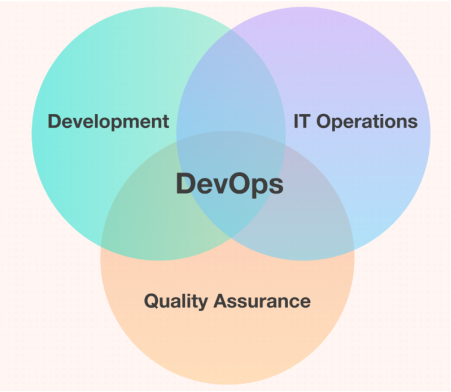
I will be documenting all the learnings from the DevOps Pro course by Hitesh Choudhary.

# ****\_What is DevOps ?****

It is a cultural practice in an organization by development team and operations team to use each other’s tools, to smooth out the process of software delivery.

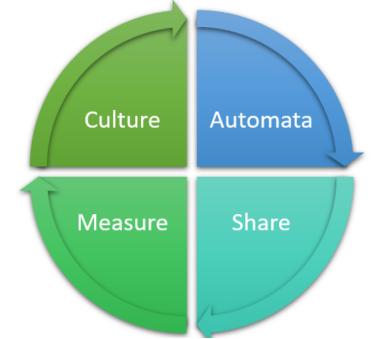


DevOps is a work culture primarily centered around collaboration, communication, and integration among the development teams. It was introduced to address the disconnect primarily between the development, operations, and ****quality assurance**** teams.



# ****\_DevOps Principles - The CAMS Mode**l**

CAMS stands for Culture, Automation, Measurement, and Sharing.



****1. Culture****

People > Process > Tools

Culture is defined by the interaction of people and groups and is driven by behavior of the people. They should talk to each other. Culture will improve if there is mutual understanding of others and their goals, interests and responsibilities.

****2. Automation****

Automation saves time, effort. It reduces the human intervention. It also focuses on people and processes and not just tools like Culture.

****3. Measurement****

Measure to improve.

Measurement is all about monitoring and keeping a track of all the progress of various activities involved in the DevOps process.

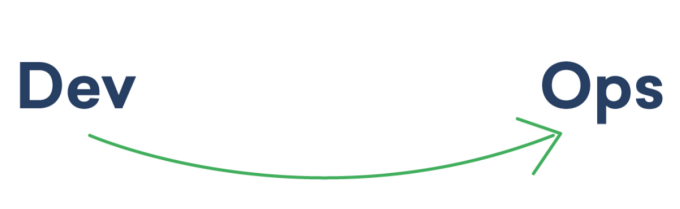
****4. Sharing****

Sharing is important in DevOps. It helps in continuous improvement of the process. Sharing the tools, knowledge and discoveries among others helps in fulfilling the end goal in a more efficient way.

# ****\_DevOps - The 3 ways****

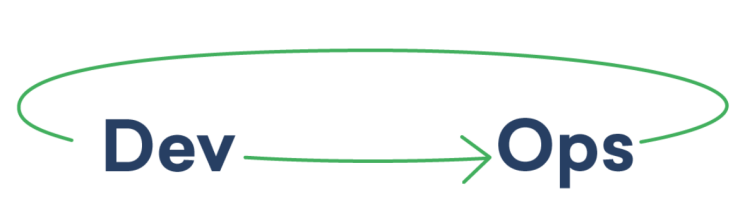
****1. Systems Thinking****

Understand and increase flow.



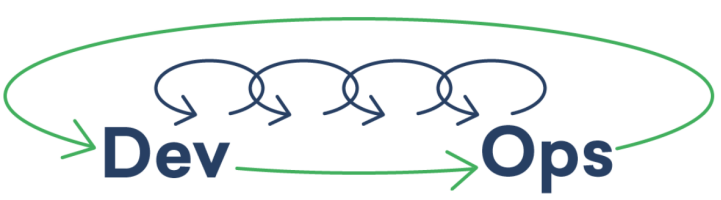
****2. Amplify Feedback****

Shorten and amplify feedback loops.



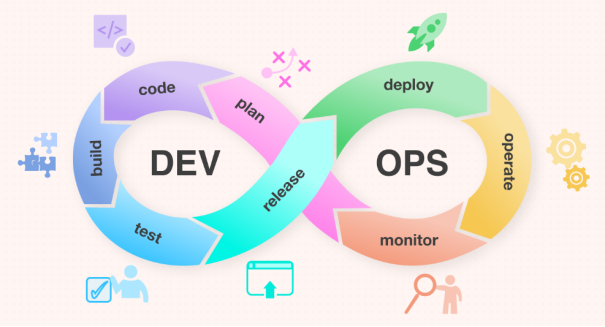
****3. Experiment and Learn****

Continual experimentation, taking risks and learning from failure.



# ****\_DevOps Lifecycle****

****Overview how the DevOps lifecycle****



1. ****Plan:****In this stage, teams identify the business requirement and collect end-user feedback. They create a project roadmap to maximize the business value and deliver the desired product during this stage.
2. ****Code:**** The code development takes place at this stage. The development teams use some tools and plugins like *Git* to streamline the development process, which helps them avoid security flaws and lousy coding practices.
3. ****Build:****In this stage, once developers finish their task, they commit the code to the shared code repository using build tools like Maven and Gradle.
4. ****Test:****Once the build is ready, it is deployed to the test environment first to perform several types of testing like user acceptance test, security test, integration testing, performance testing, etc., using tools like JUnit, Selenium, etc., to ensure software quality.
5. ****Release:**** The build is ready to deploy on the production environment at this phase. Once the build passes all tests, the operations team schedules the releases or deploys multiple releases to production, depending on the organizational needs.
6. ****Deploy:**** In this stage, Infrastructure-as-Code helps build the production environment and then releases the build with the help of different tools.
7. ****Operate:****The release is live now to use by customers. The operations team at this stage takes care of server configuring and provisioning using tools like Chef.
8. ****Monitor:****In this stage, the DevOps pipeline is monitored based on data collected from customer behavior, application performance, etc. Monitoring the entire environment helps teams find the bottlenecks impacting the development and operations teams’ productivity.

# ****\_DevOps Jargons****

****1. Provisioning****

Making sure the server is ready with operating system, software and networking is provisioning.

****2. Deployment****

Adding or upgrading the software on the server. Publishing.

****3. Orchestration****

Co-ordinated operations on multiple systems.

****4. Configuration Management****

Managing server configuration via files such as RAM, space, dependency, software, etc.

****5. Imperative Method****

Using commands to produce desired result.

****6. Declarative Method****

Here desired state is defined and tools will have to achieve it.

****7. Idempotent****

Repeat the execution and same result is given. No duplication of the result.

****8. Blue Green Deployment****

Identical deployment can be done on more than one server and it can be used as switch.

****9. Continuous Integratio****n

Building and unit testing stages of the software release process.

****10. Continuous Delivery****

Deploying on production live environment at every new check in.

****11. Continuous Deployment****

After unit testing deploy changes to production in small batches.

\_ Thank you for reading!

\_Rajani