# What is DevOps & Why DevOps?

>> DevOps is a methodology aims to close the gap between the software Development and IT operations teams in order to improve the collaboration, efficiency, delivery high quality software faster

>> It is not a single tool or framework

|  |  |  |
| --- | --- | --- |
| Aspect | Before DevOps | After DevOps |
| Collaboration | Dev & Ops work separately | Dev & Ops work together |
| Code Integration | Manual, slow integration | Continuous Integration (CI) |
| Testing | Done at the end | Automated & continuous testing |
| Deployment | Manual | Automated with Continuous Deployment (CD) |
| Software Releases | Every few weeks/months | Multiple releases per day |

# SDLC

SDLC (Software Development Life Cycle) is the stages involved in the development of software.

A diagram of software development

Description automatically generated

#### 

**Requirement Analysis**

Objective: To collect and document the needs or requirements for the software product.

**Software Design: design is done by software architects**

Process of defining the architecture, components, interfaces, and data flow of a software system to meet specified requirements,

**Software Build: done by devlopers**

* Objective: To build the actual software based on the design and requirements.
* Writing the code according
* Using version control systems (like Git) to manage the codebase.

**Testing: done by QA engineers**

* Objective: To identify defects and verify that the software meets the requirements.
* Ensure that the software is free from bugs and errors.

**Deployment**:

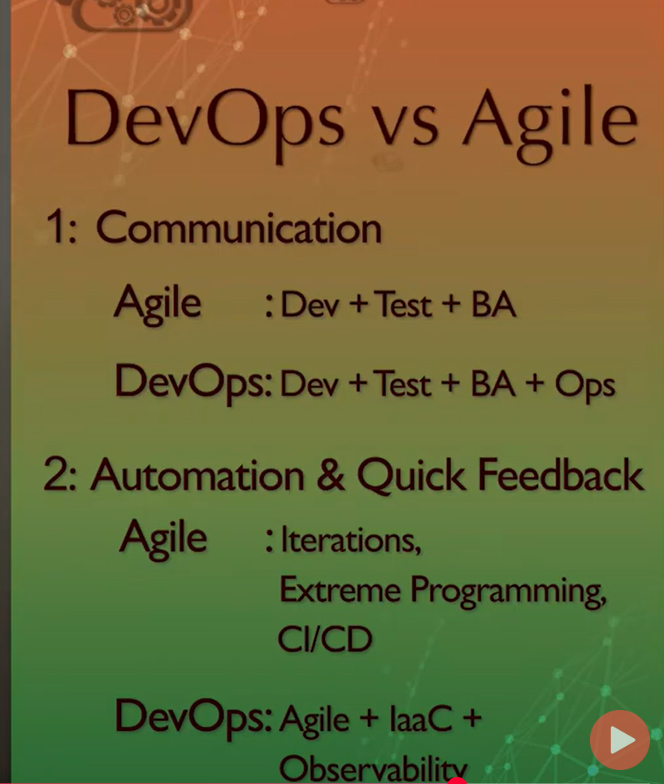
* Objective: To deploy the software into the production environment.
* Release the software to end users

**Maintenance and Support:**

* Objective: To provide ongoing support and fix any issues that arise post-deployment.

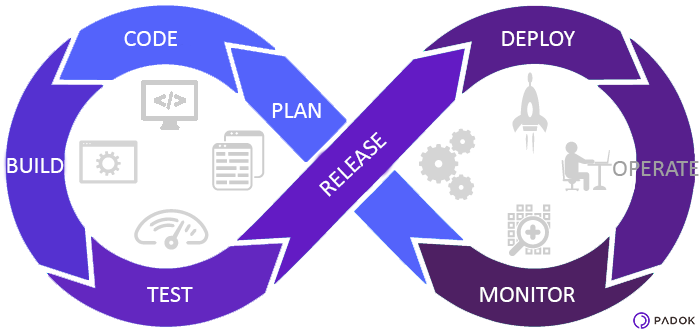
**Primarily devops engineer focus on Build,Testing and Deployment.**

Agile Vs DevOps



IAC: **Infrastructure as Code**.

# DevOps Flow Chart:



**Plan**: Defining the requirements

**Code**: Writing code based on the requirements

**Build**: compiling the code, The process of converting human-readable code (like Java, C++, etc.) into machine-readable instructions (binary or bytecode) that can be executed by the operating system.

**Test**: In the testing phase, the code undergoes rigorous testing to ensure its quality and functionality.

**Release**: Preparing the software to be available to users, Making the software ready for public access

**Deploy**: the deployment phase is about actually installing the software on the production environment and making it available for end-users

**Operate**: In this phase, the application is running in the production environment,ensuring that the application is running smoothly in the **production environment**

**Monitor**: Continuous monitoring provides feedback on the performance of the software and any potential issues.

Real-Time Monitoring: **CPU usage**, **memory consumption**, **disk I/O**, **network latency**, **response time**, etc

Logging: capturing **events**, **errors**, and **system actions**.

Security Monitoring:Security monitoring helps identify potential vulnerabilities, breaches, or unauthorized access attempts.

Feedback Loop:

After monitoring, the feedback gathered is used to plan the next iteration, and this cycle repeats. The process continues to improve the software, fix bugs, and release new features, ensuring the system evolves to meet user needs and business goals.

The 7 phases of the **DevOps lifecycle**

**PLAN,CODE → Continuous Development**

**BUILD → Continuous Integration**

**TEST → Continuous Testing**

**RELEASE, DEPLOY → Continuous Deployment**

**OPERATE AND MONITOR → Continuous Monitoring & Continuous Operation**

Phase Tools

Continuous Development: Git, GitHub, GitLab, Visual Studio Code, IntelliJ IDEA, JIRA, Trello, Confluence

Continuous Integration: Jenkins, CircleCI, GitLab CI, Travis CI, Maven, Gradle, JUnit, Selenium, SonarQube

Continuous Testing: JUnit, Selenium, Cucumber, Postman, JMeter, Gatling, OWASP ZAP, JaCoCo, Istanbul

Continuous Deployment: Jenkins, GitLab CI/CD, CircleCI, Docker, Kubernetes, Helm, AWS CodeDeploy, Google Cloud Deployment

Continuous Monitoring: Prometheus, Nagios, Zabbix, New Relic, Datadog, AppDynamics, ELK Stack, Splunk, Fluentd, Sentry

Continuous Operations: Ansible, Chef, Puppet, Terraform, CloudFormation, PagerDuty, Opsgenie, Kubernetes, AWS Auto Scaling