**WHAT IS DEVOPS**

**DevOps** is a combination of **Development (Dev)** and **Operations (Ops)** — a **culture, process, and set of tools** that bring together software development and IT operations teams to deliver applications **faster, more reliably, and continuously**.

**Key Idea:**

DevOps aims to **break the barrier** between developers who write code and operations teams who deploy and maintain it — promoting **collaboration, automation, and continuous improvement**.

**Main Goals:**

* **Faster software delivery**
* **Improved deployment frequency**
* **Early detection and quick fixing of issues**
* **Better product quality and reliability**

**Core Practices:**

1. **Continuous Integration (CI)** – frequently merging code into a shared repository and testing it automatically.
2. **Continuous Delivery (CD)** – automatically deploying code to production or staging environments.
3. **Infrastructure as Code (IaC)** – managing infrastructure using code (e.g., Terraform, Ansible).
4. **Monitoring & Logging** – tracking performance and system health in real time.
5. **Collaboration & Communication** – using shared tools and agile methods.

**Common DevOps Tools:**

| **Stage** | **Tools** |
| --- | --- |
| Version Control | Git, GitHub, GitLab |
| CI/CD | Jenkins, GitHub Actions, Azure DevOps |
| Configuration Management | Ansible, Puppet, Chef |
| Containerization | Docker |
| Orchestration | Kubernetes |
| Monitoring | Prometheus, Grafana, ELK Stack |

**Benefits:**

* Faster delivery of features
* Fewer deployment failures
* Quicker recovery from errors
* Better teamwork between Dev and Ops

**Main advantages of DevOps:**

**1. Faster Software Delivery**

Automation and CI/CD pipelines help release new features and updates **quickly and frequently**.

**2. Improved Collaboration**

Bridges the gap between **development and operations teams**, promoting shared responsibility and better communication.

**3. Higher Deployment Success Rate**

Automated testing and consistent environments reduce **bugs and deployment failures**.

**4. Better Quality and Reliability**

Continuous monitoring and feedback loops ensure **early issue detection** and stable software performance.

**5. Increased Efficiency**

Automating repetitive tasks like testing, deployment, and configuration saves **time and manual effort**.

**6. Scalability and Flexibility**

Using tools like **Docker** and **Kubernetes**, applications can be **scaled easily** to handle demand.

**7. Continuous Feedback and Improvement**

Real-time monitoring provides feedback that helps teams **improve code, infrastructure, and user experience**.

**8. Faster Recovery from Failures**

In case of errors, DevOps enables **quick rollback** and faster system recovery.

**9. Cost Efficiency**

By reducing downtime and manual intervention, organizations **save time, resources, and costs**.

**WHAT IS AGILE**

Agile is a software development methodology that focuses on flexibility, collaboration, and continuous improvement to deliver software quickly and efficiently.

**Key Idea:**

Instead of building the whole product at once, Agile breaks the project into small, manageable parts called iterations or sprints, each delivering a working feature.

Core Principles (from the Agile Manifesto):

1. Customer satisfaction through early and continuous delivery.
2. Welcome changing requirements, even late in development.
3. Frequent delivery of working software.
4. Close collaboration between business and developers.
5. Motivated teams with trust and support.
6. Face-to-face communication as the best form of interaction.
7. Working software is the main measure of progress.
8. Sustainable pace — development should be steady and manageable.
9. Technical excellence and good design enhance agility.
10. Simplicity — focus only on what’s needed.
11. Self-organizing teams produce the best results.
12. Regular reflection for continuous improvement.

**Popular Agile Frameworks:**

* Scrum – uses sprints, daily standups, and roles like Product Owner, Scrum Master, and Development Team.
* Kanban – focuses on continuous flow using boards and cards to visualize tasks.
* Extreme Programming (XP) – emphasizes frequent releases and customer involvement.

**Benefits:**

* Faster delivery of value
* Better adaptability to changes
* Improved customer satisfaction
* Continuous feedback and improvement
* Enhanced team collaboration