DevOps

What is Waterfall Methodology?

The **Waterfall model** is a **sequential, linear approach** to software development where progress flows in one direction—like a **waterfall**—through distinct phases.

Phases of the Waterfall Model:

1. **Requirement Gathering** – All requirements are collected before development starts.
2. **Design** – System architecture and technical design are created.
3. **Implementation (Development)** – Developers write code based on the design.
4. **Testing** – The software is tested for defects and issues.
5. **Deployment** – The product is released to users.
6. **Maintenance** – Fixes and updates are made as needed.

Royce in 1970

What is Agile Methodology?

Agile is a **software development approach** that emphasizes **iterative progress, collaboration, and flexibility**. It focuses on delivering small, functional increments of a project rather than a single, large release.

The Agile methodology, as defined by the "Agile Manifesto," was not invented by a single person, but rather developed by a group of software developers

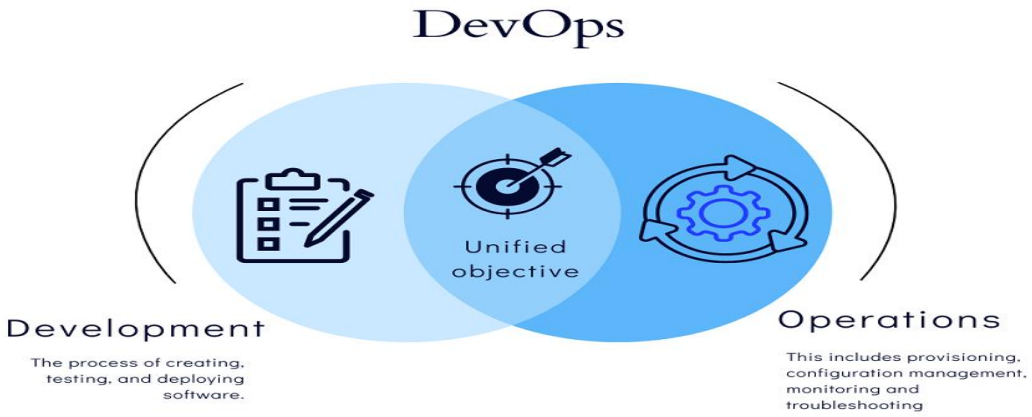
including Kent Beck, Martin Fowler, Ron Jeffries, Ken Schwaber, and Jeff Sutherland who came together in 2001

How Agile Works:

- Development is broken into **small iterations (sprints)**, usually lasting 1-4 weeks.
- Teams **continuously test, review, and improve** the product.
- **Customer feedback** is integrated at each stage to ensure alignment with business needs.
- Agile embraces **changes and improvements**, unlike traditional waterfall models.

What is DevOps?

DEV + OPS



DevOps is a set of practices, tools, and cultural philosophies that aim to improve collaboration between development (Dev) and operations (Ops) teams. The goal is to shorten the software development lifecycle, ensuring faster and more reliable software releases.

Key Aspects of DevOps:

1. **Collaboration & Communication:** Breaks silos between development and IT operations.
2. **Automation:** Automates processes like testing, integration, and deployment.
3. **CI/CD (Continuous Integration & Continuous Deployment):** Ensures frequent code updates with minimal manual intervention.
4. **Infrastructure as Code (IaC):** Manages infrastructure through code (e.g., Terraform, Ansible).
5. **Monitoring & Feedback:** Uses tools like Prometheus, Grafana, and ELK stack for real-time insights.
6. **Scalability & Security:** Ensures applications are scalable and secure from development to deployment.

Popular DevOps Tools:

- **CI/CD:** Jenkins, GitHub Actions, GitLab CI/CD
- **Configuration Management:** Ansible
- **Containerization:** Docker, Kubernetes
- **Infrastructure as Code (IaC):** Terraform
- **Monitoring & Logging:** Prometheus, ELK Stack (Elasticsearch)
- **Cloud Platforms:** AWS.

How does DevOps work?

- **Continuous integration (CI)**: Automates the process of merging code changes from multiple contributors into a single project
- **Continuous delivery (CD)**: Automatically packages and deploys code modifications to production
- **Automation**: Includes monitoring, reporting, testing, and deployment

What is a Server?

A **server** is a computer or software system that provides services, resources, or data to other computers (called **clients**) over a network. Servers are used in various environments, including web hosting, databases, file storage, and application deployment.

Types of Servers

1. **Web Server** – Hosts websites (e.g., Apache, Nginx).
2. **Application Server** – Runs application logic (e.g., Tomcat, JBoss).
3. **Database Server** – Manages databases (e.g., MySQL, PostgreSQL, Oracle).

What is a Service?

A **service** is a process or function that runs in the background on a computer or a network, providing specific capabilities to users or other systems. Services can be software-based (like web servers, databases) or infrastructure-based (like cloud computing services).

Running an index.html File in an Nginx Server

sudo apt update

sudo apt-get install nginx

nginx -v

sudo systemctl status nginx

cd /var/ www/html

vim index.html