



COMP60009 – Emerging Technologies 1 – Lecture 1

VIJU PRAKASH

Senior Lecturer,

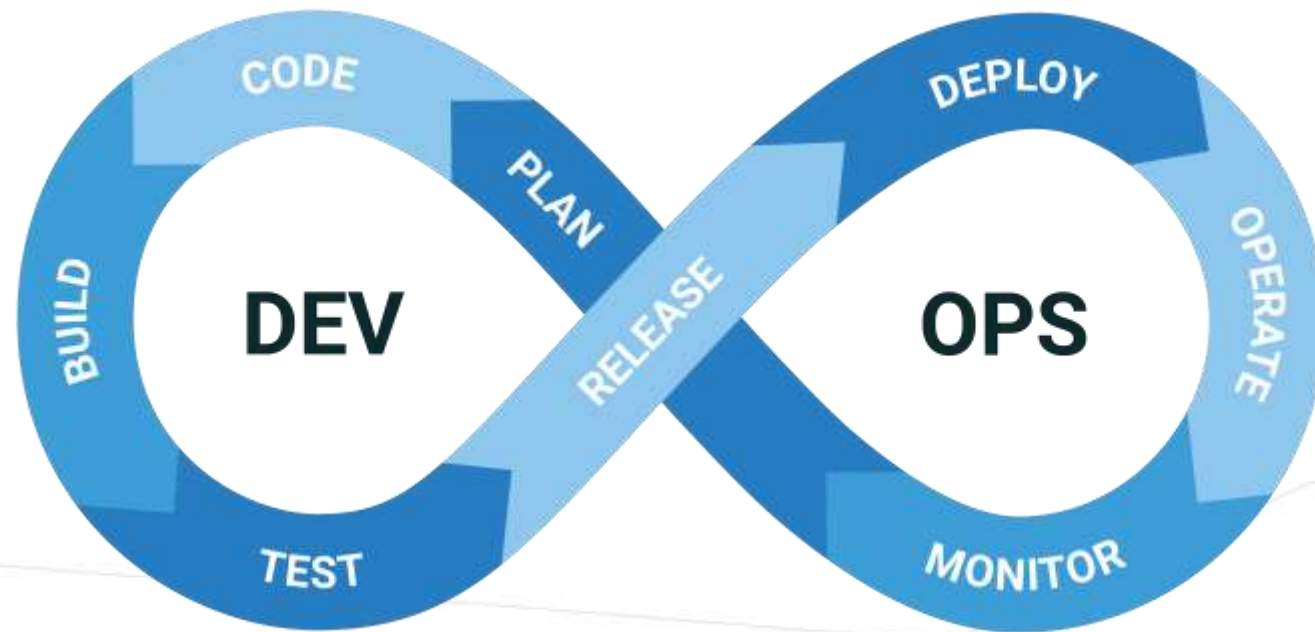
Department of Computer Science,

BUV, Hanoi.



www.buv.edu.vn

INTRODUCTION TO DEVOPS



- **DevOps** is not a technology, it's a **culture**.
- **Development + Operations**
- Imagine that you are working for a company. There are a few developers (programmers) in your company. They are using some programming languages to develop the project.

- One of your clients needs to design an e-commerce website.

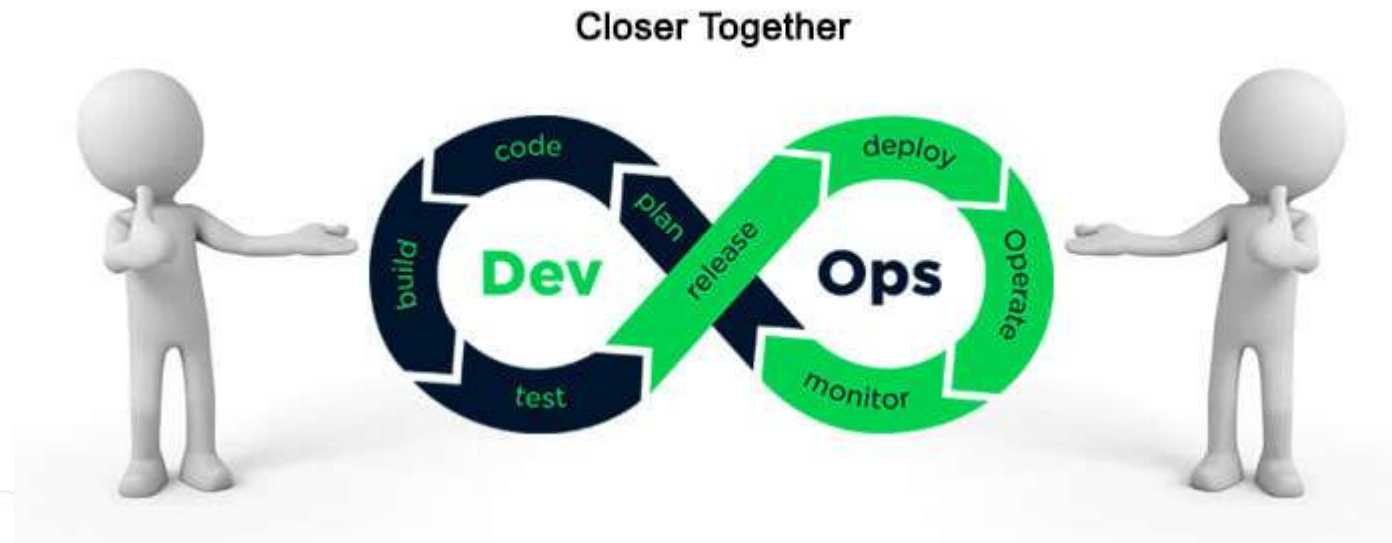


- Can we give the code, or the project developed by the programmers to the clients directly? If you say no, then say the reason.
- The roles of the development team are
 1. Understand the client's requirements.
 2. Based on the requirements, they develop the programs.
 3. Test the programs.
 4. Keep code in a repository (GitHub).

- What are the roles of the IT operations team?

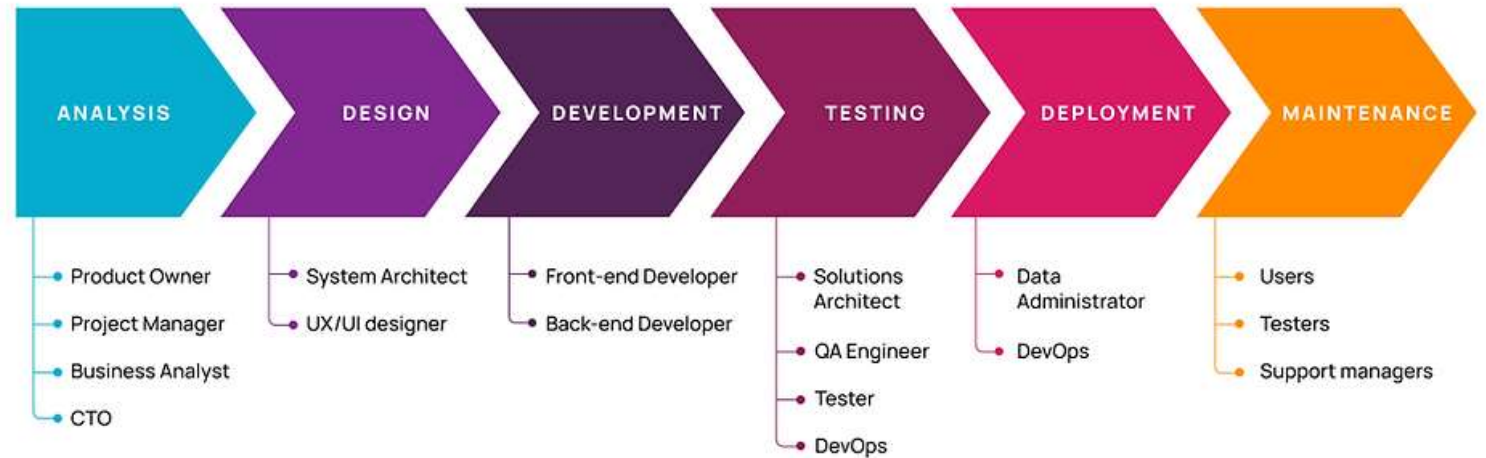
1. Setup virtual machines
2. Setup servers
3. Setup database
4. Take code from a repository
5. Code build and package
6. Deploy code in server
7. Deliver the project to the client.

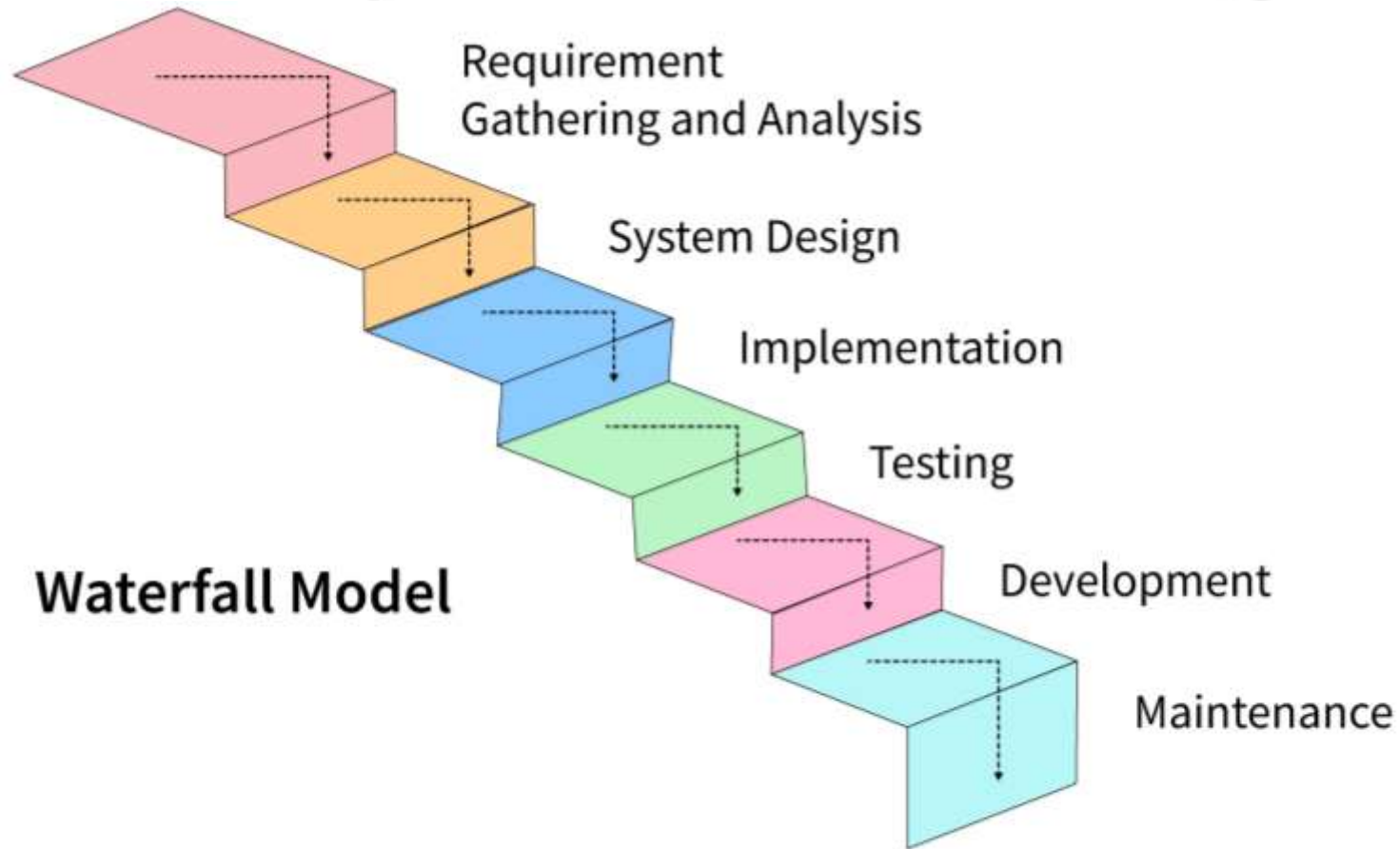
- You may understand now DevOps is not a single person's job. It's a teamwork.
- Developers and IT Operations team have to work together



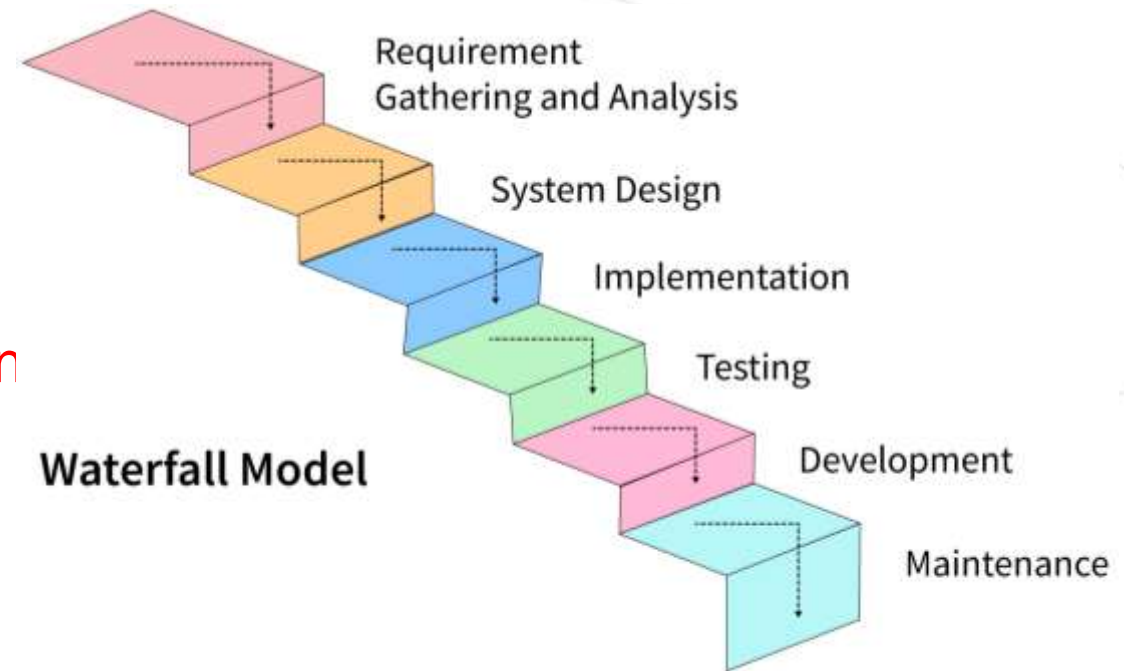
- Speed
- Improved collaboration
- Rapid delivery
- Quality and reliability

- Requirements Gathering
- Requirements Analysis
- Planning or Design
- Coding or Implementation
- Testing
- Deploy the project
- Deliver the project
- Maintenance





- Linear model
- Requirement change is difficult
- The client should wait for a long time
- Cause a huge loss if something goes wrong
- Client involvement is less





- Iterative model
- Requirement change is possible.
- Development and testing will go parallel.
- Projects will be released periodically (Sprints)
- Feedback will be collected from the clients after releasing a sprint
- Client involvement is more
- Development and testing is frequent or agile.



WHICH MODEL IS SUITABLE FOR YOU?

WATERFALL

You can accomplish things in succession, one at a time

For those with simple or more basic project goals, the waterfall methodology relies on doing things one step at a time.

You have a long lead time to accomplish your goals

If you have a lot of time, doing things step-by-step isn't a problem.

You already have a fully developed product that needs few to no changes

You have a product that doesn't rapidly change in step with market developments or new technology.

You only need results at the end of your project

If you're dealing with a long-lead project that needs few changes, there's less urgency to get results as quickly as possible.

You already have a set roadmap of how to get to what you need

Best for projects that have worked before and need little changing.

AGILE

You need to accomplish many things all at once

The agile SDLC is built for speed, allowing you to complete multiple stages of your project simultaneously.

You have a tight deadline to meet your project goals

The agile SDLC is built to consolidate and quickly move towards all of your project goals at once, rather than one at a time.

You have a basic product that needs to go through multiple iterations of improvement

If your product is in a basic form, the agile SDLC helps improve it through fast-evolving prototypes, giving you the best final product in the least amount of time.

You need ongoing results delivered to you throughout the project

For projects going through multiple iterations, get multiple results on iterations before reaching a final product.

You are working with a new product and need fluidity

The agile SDLC is great for projects that are brand new, as it allows for fluid movements between what works and what doesn't.

- Build tools – (Maven, Gradle)
- Repository tools – (SVN, GitHub, BitBucket)
- Code review tools – (Sonar Qube, Sonar lint)
- Code deployment tools (Jenkins, Udeploy)
- Configuration tools (Chef, Ansible)
- Containerization tools (Docker)
- Orchestration tools (K8S)
- Monitoring tools (Nagios, Grafana, Prometheus, Splunk)
- Project Management (Jira)

- It is a build tool
- Open source given by Apache
- It is developed by Java
- It is used to build automation for Java projects.
- Maven is called Java Build Tool

- We can create an initial project folder structure.
- We can download project dependencies.
- Required dependencies should be mentioned in pom.xml
- When we create a maven project, pom.xml will be created.
- We can compile project source code using Maven.
- We can pack a Java project as a jar (Java archive) or war (Web archive) file.
- Standalone Java applications are executed using jar files.
- Java web applications are executed using war files.

- Download and install Java software.
- After installing Java, you will get JDK and JRE.
- JDK contains a set of tools to develop Java programs.
- JRE contains the platform/environment that is used to run the Java programs.
- Once the installation is done, set JAVA_HOME in the system environment variables.
- Then set path for java (bin file)
- Verify Java execution by using `java -version` command.
- Now download Maven software from Apache website. Extract and paste it in C drive.
- Set MAVEN_HOME in System environment variable.
- Then set path for Maven (bin file)
- Verify Maven execution by using `mvn -version` command.



THANK YOU

BUV Ecopark Campus, Ecopark Township, Van Giang, Hung Yen

 www.buv.edu.vn

 info@buv.edu.vn