

# Thadomal Shahani Engineering College

Bandra (W.), Mumbai - 400 050.

## © CERTIFICATE ©

Certify that Mr./Miss Yatish Shah  
of AI&DS Department, Semester V1 with  
Roll No. 91 has completed a course of the necessary  
experiments in the subject SEPM under my  
supervision in the **Thadomal Shahani Engineering College**  
Laboratory in the year 2024 - 2025

Teacher In- Charge

Head of the Department

Date 14/03/25

Principal

## SEPM Assignment 2.

Aim: To understand Devops: Principles, Practices & Devops Engineer Role & responsibility.

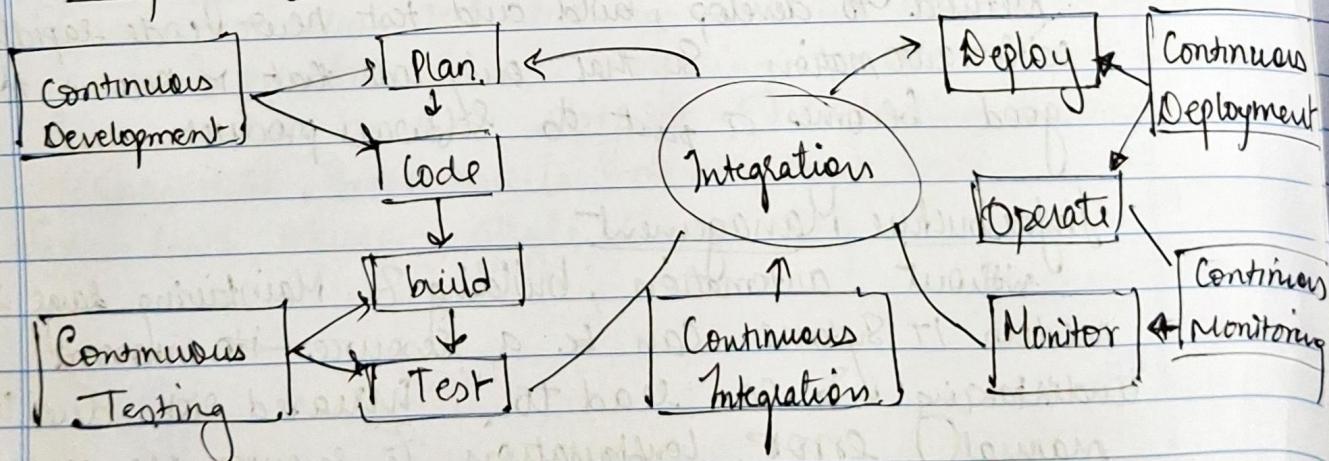
### What is Devops?

Devops is a collaborative approach where teams work together to build & deliver secure software efficiently. It consists of Software Development (Dev) & Operations (Ops) to decide how to accelerate delivery through automation, collaboration, fast feedback, & iterative improvements. Built on Agile methodology. Devops creates a culture of collaboration & shared responsibilities for business outcomes.

### Core principle :-

1. Develop and test in production like environment
2. Deploy builds frequently.
3. Continuously validate Operational Quality.

### Devops Practices:



## Continuous Integrations

Continuous Integration refers to the build & unit testing stages of the software releases process. Every revision that is committed triggers an automated build & test.

Ex: Jenkins, Travis, Circle.

## Continuous Delivery & Deployment

This originates from Continuous Integration, a method to develop, build and test new code rapidly with automation so that only code that is known to be good becomes a part of software product.

## Infrastructure Management

Without automation, building & maintaining large scale modern IT systems can be a resource-intensive undertaking & can lead to increased risk due to manual error, configuration & resource management.

# Configuration Management

Infrastructure as Code is the practice of describing all software runtime environment & networking settings & parameters in simple textual format, that can be stored in your Version Control System (VCS) & versioned on request. These text files are called as Manifest & are used by DevOps tools to automatically provision & configure servers.

## Microservice Architecture:-

Docker is a tool designed to make it easier to create, deploy, & run applications by using containers. Containers allows a developer to package up an application with all of the parts it needs, such as libraries & other dependencies & deploy it as one package. By doing so, thanks to the containers the developer can rest assured that the application will run on any other Linux machine.  
Ex Nagios, Splunk etc.

## Cloud Based DevOps

DevOps automation is becoming Cloud Centric. Most Public & private Cloud computing providers support DevOps systematically on their platform, including continuous integration & continuous development tools.  
Ex AWS, Azure, google cloud etc.

DevOps engineer

A DevOps Engineer manages a company's IT infrastructure, development & operations. Key responsibilities include:-

### Technical Responsibilities:

- Implement development, testing & automation tools.
- Setup infrastructure & tools.
- Code review & responsibilities
- Bug fixing & troubleshooting
- Security implementations & monitoring

### Management Responsibilities

- Understand customer requirements & APIs.
- Plan team structure & activities
- Manage stakeholders
- Coordinate team communication
- Monitor customer experience.
- Provide periodic progress reports
- Mentor team members
- Define development & operational process.

Yatish Shah

14/11/2025

## Assignment - 1

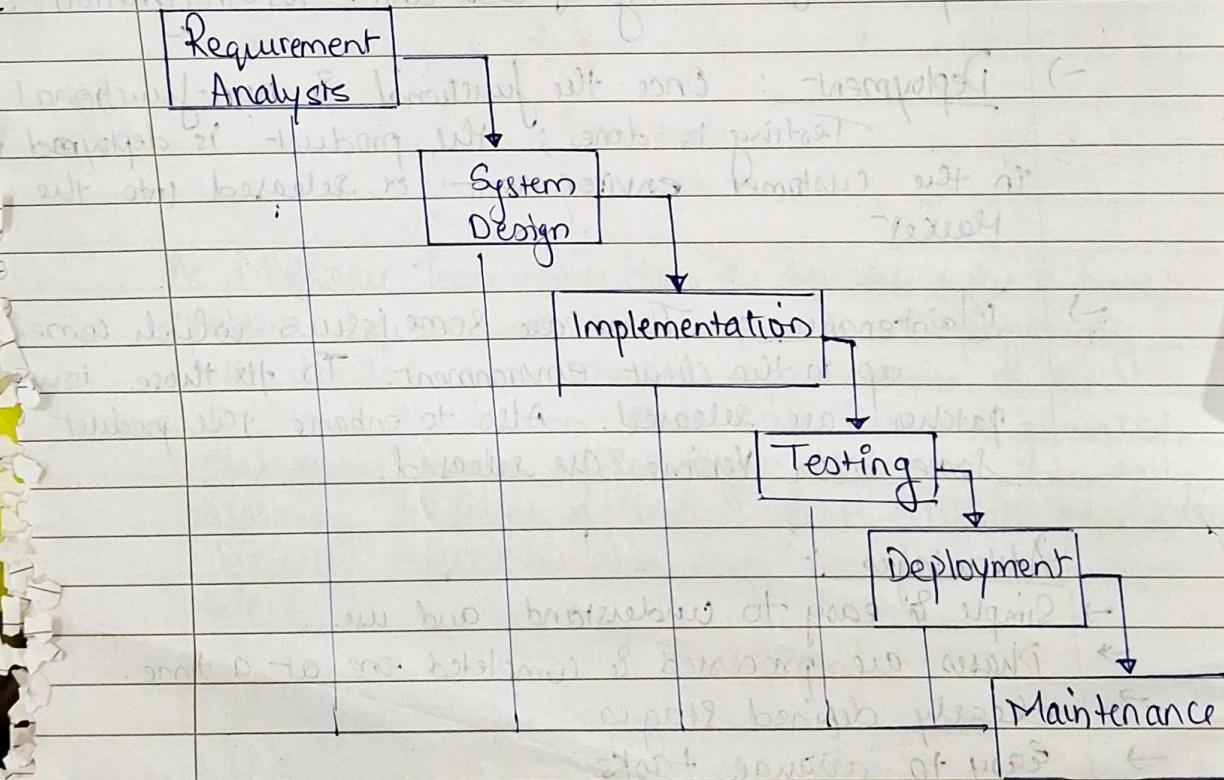
### The Waterfall Model.

In This Model the requirements for a problem are well-understood - when work flows from Communication through deployment in a reasonably linear fashion

The Waterfall Model, sometimes called the

Classic life cycle, suggests a systematic, sequential approach to software development that begins

with customer specification of requirements & progresses through planning, Modeling, construction and deployment, culminating in ongoing support of the completed software.

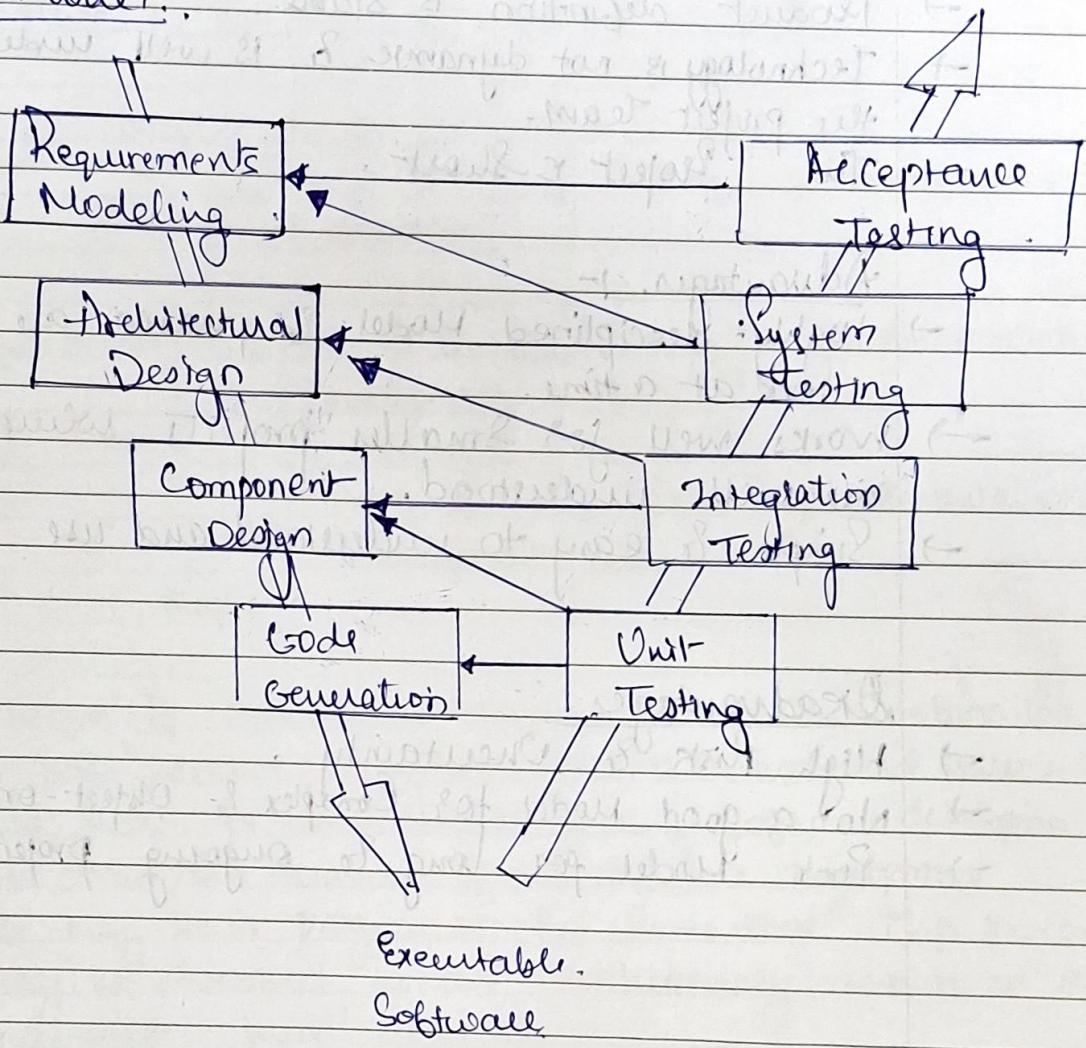


- Requirement Gathering & Analysis : All possible requirements of the system to be developed are captured in this phase & documented in a requirement specification doc.
- System Design : The requirement specifications from first phase are studied & system design is prepared, which helps specify hardware & architecture.
- Implementation :- With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase.
- Integration & Testing : All the units developed in the implementation phase are integrated into a system after testing of each unit. Post Integration.
- Deployment : Once the functional & non-functional testing is done, the product is deployed in the customer environment or released into the market.
- Maintenance : There are some issues which come up in the client environment. To fix those issues patches are released. Also to enhance the product some better versions are released.

#### Advantages :-

- Simple & easy to understand and use.
- Phases are processed & completed one at a time.
- Clearly defined stages.
- Easy to arrange tasks.

## V-Model



As a Software Team moves down the left side of the V, basic problem requirements are defined into progressively more detailed and technical representations of the problem & its solution. Once code has been generated, the team moves up the right side of V, essentially performing a series of tests that validate each of the models created as the team moved down the left side.

Applications:-

- Product definition is stable.
- Technology is not dynamic & is well understood by the project team.
- The Project is short.

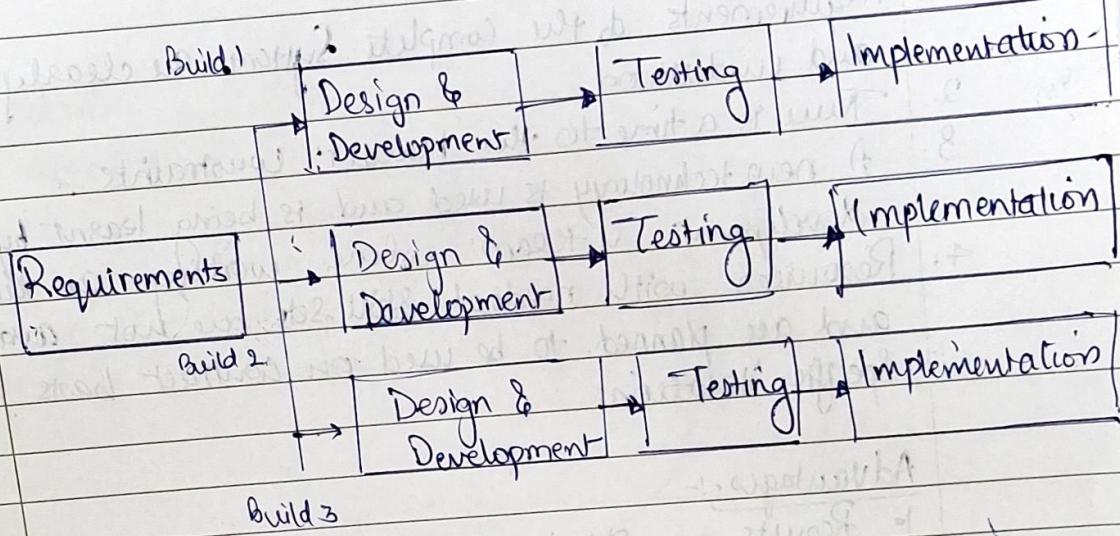
### Advantages:-

- Highly disciplined Model & phases are completed one at a time.
- works well for smaller projects where requirements are well understood.
- Simple & easy to understand and use.

### Disadvantages :-

- High Risk & Uncertainty.
- Not a good model for Complex & Object-oriented projects.
- Poor Model for long & ongoing projects.

## Iterative Model & Incremental Model



Iterative & Incremental development is a combination of both iterative design or iterative method and incremental build model for development. During software development more than one iteration of the software development cycle may be in progress at the same time. This process may be described as an "evolutionary acquisition" or "incremental build" approach.

In this Incremental Model, the whole requirement is divided into various builds. During each iteration, the development module goes through the requirements, design, implementation & testing phases. The key to a successful use of an iterative software development lifecycle is rigorous validation of requirements, and verification & testing of each version of the software against those requirements. Within each cycle of the model,

## Applications:-

1. Requirements of the complete system are clearly defined and understood.
2. Time is a constraint.
3. A new technology is used and is being learnt by the development team while working on the project.
4. Resources with needed skill sets are not available and are planned to be used on contract basis for specific iterations.

## Advantages:-

1. Results are obtained early & periodically.
2. Progress can be measured.
3. less costly to change the scope/ requirements.
4. Testing & debugging during smaller iterations is easy.

## Disadvantages:-

1. More Resources may be required.
2. More Management attention is required.
3. Not suitable for small projects.
4. Management complexity is more.

## Spiral Model

Objective  
Identification.

Progress

Alternate  
Evaluation.

Review

next phase  
Planning.

Product  
Release

Development

- ① Identification :-  
This phase starts with gathering business requirements in the baseline spiral. In the subsequent spirals as the product matures, identification of system requirements is done in this phase.

- ② Design :-  
Starts with conceptual design in the baseline spiral & involves architectural design, logical design of modules,

- ③ Construct :-  
The construct phase refers to production of the actual software product at every spiral. In the baseline spiral, when the product is just thought of & the

## Evaluation & Risk Analysis

Risk Analysis includes identifying, estimating & monitoring the technical feasibility & management risks, such as schedule slippage and cost overruns.

### Advantages:

- changing Requirements can be accommodated.
- Allows extensive use of prototypes.
- Users see the System early.
- Requirements can be captured more accurately.

### Disadvantages:

- Management is more complex.
- End of project may not be known early.
- Spiral may go on indefinitely.
- process is complex.