

# Assignment 1

Name: UTKARSH BOTHE

PRN: 22311823

ROLL NO: 62

Subject: Software Analysis and Design

**Title:** To study software process modeling methodologies and identify their applicability to various categories of projects

## 1. Introduction

When we make software, we cannot just start coding directly. There has to be a proper process to understand the problem, design the solution, write the code, test it, and finally deliver it. This step-by-step approach is called a Software Process Model.

Different models exist because not all projects are the same. A small college project, a banking system, and an aerospace control system all need different approaches. Some projects have fixed requirements, while others keep changing. That's why choosing the correct process model is very important.

## 2. Software Process Modeling Methodologies

### a) Waterfall Model

This is the oldest and most traditional way of developing software. In this model, the work is done step by step, like water flowing down a waterfall. Once you finish one step, you move to the next. The phases are: Requirements → Design → Coding → Testing → Deployment → Maintenance.

- Where to use?
  - Best for projects where requirements do not change.
  - For example: billing systems, library management, government software where rules are fixed.
- Limitations: If the customer changes his mind in the middle, it is very difficult to go back.

### b) Incremental Model

Instead of building the whole software at once, here we build it in small parts (increments). Each part adds more features. For example, first release may have login, second release may add payment, third release may add reports.

- Where to use?
  - Useful when requirements are big and complex.
  - If the customer wants to see some working software early.
  - For example: Banking software, E-commerce websites.
- Limitations: Needs good planning, otherwise parts may not fit well together.

### **c) Spiral Model**

This model looks like a spiral because the project moves in repeated cycles. Each cycle has four main steps: planning, risk analysis, development, and evaluation. After one cycle, the spiral continues with improvements.

- Where to use?
  - For very large projects where risks are high.
  - For example: Space research systems, air traffic control software, medical devices.
- Limitations: It is costly because risk analysis takes a lot of time and experts.

### **d) V-Model (Verification & Validation Model)**

The V-model is like the waterfall model but with more focus on testing. For every development stage, there is a testing stage planned side by side. For example, requirements have acceptance testing, design has integration testing, coding has unit testing.

- Where to use?
  - Projects where high quality and accuracy are needed.
  - For example: Car safety systems, airplane software, hospital equipment.
- Limitations: Same as waterfall – not suitable when requirements keep changing.

### **e) Agile Methodology**

Agile is one of the most popular modern approaches. Instead of making the whole software at once, Agile focuses on small iterations (called sprints). Developers and customers work together closely, and after every sprint, the customer gets to see working software and give feedback.

- Where to use?
  - Projects where requirements are not fixed and may change frequently.
  - For example: Mobile apps, web apps, startup projects.
- Limitations: Needs active involvement of customer and skilled team members.

## f) Prototype Model

Here, before making the actual software, a working model (prototype) is made. This helps the customer to see how the system will look and behave. Based on feedback, the prototype is improved until requirements are clear. After that, the actual system is developed.

- Where to use?
  - When requirements are not clear at the beginning.
  - For example: New websites, UI/UX heavy apps, games.
- Limitations: Making too many prototypes can waste time and money.

## 3. Comparative Table

Model	Best For	Example
Waterfall	Small, stable projects	Library management system
Incremental	Large projects with gradual delivery	Banking system
Spiral	High-risk, critical projects	Space/aerospace software
V-Model	Safety-critical systems	Medical or automotive systems
Agile	Fast-changing projects	Mobile apps, startups
Prototype	Projects with unclear needs	E-commerce website design

## 4. Conclusion

No single model is perfect for all types of projects. Each methodology has its own strengths and weaknesses. If requirements are stable, Waterfall or V-Model is good. If requirements keep changing, Agile or Incremental works better. If the project is high-risk, Spiral is suitable, and if requirements are unclear, the Prototype Model is best.

The key is to understand the type of project first and then choose the process model accordingly.