

## Lab 3:

### Lab Exercise: Software Development Models

#### Objective:

To understand the features, applications, and differences between various software development models: Waterfall, Spiral, V-Model, and Agile, and apply them to real-world scenarios.

---

### Lab\_exercise1: Identification and Mapping

---

#### Instructions:

Match the following scenarios to the most appropriate Software Development Model. Justify your answer in 1-2 lines.

Sol:

Scenario	Model	Justification
1. A banking project where requirements are fixed and testing is required only after implementation.	Waterfall Model	The Waterfall model is suitable because requirements are well-defined and fixed at the beginning, allowing for a linear, sequential development process with testing at the end.
2. A mobile app where customer feedback plays a major role in shaping the features.	Agile Model	The Agile model is ideal for this scenario because it emphasizes iterative development and continuous customer collaboration, allowing for frequent feedback and adaptation to changing requirements.
3. A defense system requiring high reliability and where each phase requires testing against strict standards.	V-Model (Verification and Validation Model)	The V-Model is best suited for this project as it places a strong emphasis on verification and validation (testing) activities for each development phase, which is critical for a high-reliability system.
4. A satellite launch software with high risk and evolving requirements.	Spiral Model	The Spiral model is appropriate here due to its focus on risk management and its iterative nature, which allows for repeated cycles of planning, risk analysis, engineering, and evaluation, accommodating evolving requirements in a high-risk environment.

---

## Lab\_exercise2: Model Breakdown Analysis

---

### Instructions:

Fill in the table with **two pros** and **two cons** of each model.

Sol:

Model	Pros (2)	Cons (2)
Waterfall	<ol style="list-style-type: none"><li>1. Simple and easy to understand and manage.</li><li>2. Works well when requirements are clear and fixed.</li></ol>	<ol style="list-style-type: none"><li>1. Inflexible to changes once a phase is completed.</li><li>2. Late testing may lead to costly fixes if defects are found.</li></ol>
Spiral	<ol style="list-style-type: none"><li>1. Strong focus on risk management.</li><li>2. Suitable for large, complex, and high-risk projects.</li></ol>	<ol style="list-style-type: none"><li>1. Can be costly due to repeated iterations.</li><li>2. Requires expertise in risk assessment.</li></ol>
V-Model	<ol style="list-style-type: none"><li>1. Each development stage has a matching testing stage, ensuring high quality.</li><li>2. Good for projects requiring strict standards and reliability.</li></ol>	<ol style="list-style-type: none"><li>1. Inflexible to requirement changes.</li><li>2. Can be expensive and time-consuming for large projects.</li></ol>
Agile	<ol style="list-style-type: none"><li>1. Flexible and adaptive to changing requirements.</li><li>2. Continuous customer involvement ensures satisfaction.</li></ol>	<ol style="list-style-type: none"><li>1. Requires high customer availability and commitment.</li><li>2. Difficult to predict time, cost, and resources accurately.</li></ol>

---

## Lab\_exercise3: Project-Model Matching

---

### Instructions:

Below are four projects. Choose the most suitable model for each and explain why (2 marks for model, 1 mark for explanation):

Sol:

1. **College Management System**

Model: V-Model

Justification: A college management system requires high reliability and minimal errors, as it deals with critical data like student records and finances. The V-Model's emphasis on rigorous testing and validation at each stage ensures a high-quality, dependable system that meets all specified requirements.

2. **Netflix Clone Application**

Model: Agile Model

Justification: A media streaming application like a Netflix clone benefits from the Agile approach because user feedback and market trends are crucial for shaping features. This iterative model allows for the frequent release of new features, bug fixes, and improvements based on customer feedback, ensuring the app remains competitive and relevant.

3. **Medical Device Software**

Model: Spiral Model

Justification: Medical device software is a high-risk project where failure can have severe consequences. The Spiral Model is the best fit as it is centered around risk analysis at each stage. This iterative process allows for careful planning, risk mitigation, and prototyping, ensuring the software is safe and reliable before it's fully developed.

4. **Basic Calculator App**

Model: Waterfall Model

Justification: A basic calculator app has very fixed, well-defined requirements that are unlikely to change. The simple, sequential nature of the Waterfall Model is efficient for such a project, as development can proceed in a linear fashion from requirements gathering to testing without the need for frequent iterations or risk analysis.

---

## Lab\_exercise 4: Agile Sprint Simulation

---

### Scenario:

You are developing a **To-Do List App** using the **Agile Model**.

### Instructions:

1. Write **3 features** that will be developed.
2. Plan them over **2 sprints** (each sprint is 2 weeks).
3. Write what feedback you may expect after Sprint 1.

### Sol:

Feature	Sprint No.	Why in this Sprint?	Expected Feedback
1. User can create, edit, and delete tasks	Sprint 1	Core functionality; must be ready early so users can start testing the basic purpose of the app.	Users may suggest improvements in task editing (e.g., easier deletion, undo option) or request UI simplification.
2. Set due dates and reminders for tasks	Sprint 2	Adds value after core task management works; helps improve usability and productivity.	Users may ask for customizable reminder options (e.g., push notifications, repeat reminders).
3. Categorize tasks with tags or lists	Sprint 2	Enhances organization and is a secondary feature after the main app functions are stable.	Users may want color coding or the ability to filter/sort tasks by category.

Reference for my To-Do-List : <https://padmapriyanh.github.io/todo-list/>

Please do check it out ;)

---

## Lab\_exercise 5: Comparison Table Fill-in-the-Blanks (20 Marks)

---

Fill in the missing information based on your understanding:

Sol:

Feature	Waterfall Model	Spiral Model
Development Approach	<b>Linear</b> and sequential. Each phase must be completed before the next one begins.	<b>Iterative</b> and cyclical. The project goes through multiple spirals, with each loop representing a phase of development.
Flexibility	<b>Low.</b> Requirements are fixed at the beginning, making it difficult to incorporate changes later in the process.	<b>High.</b> It is highly adaptable and can easily accommodate changes in requirements in subsequent spirals.
Risk Handling	<b>Poor.</b> Risks are often identified too late, usually during the testing phase, making them difficult and expensive to fix.	<b>Excellent.</b> Risk analysis is a core part of each spiral, allowing for early identification and mitigation of potential risks.
Testing Phase	<b>Occurs at the end</b> of the development process, after all implementation is complete.	<b>Integrated throughout</b> the development process. Testing and validation are part of each spiral loop.
Customer Involvement	<b>Limited.</b> The customer is mainly involved at the beginning (for requirements) and at the end (for delivery).	<b>High.</b> The customer is involved in each spiral loop, providing feedback and helping to define the objectives for the next iteration.