#### **Experiment No. 2: Literature Survey on Software Development Methodology**

### **Aim:**

To conduct a literature survey on various software development models and select the most suitable model for the project based on its requirements, constraints, and desired outcomes.

### **Theory:**

### **Agile Model: Theory**

The Agile model is a modern software development methodology focused on flexibility, incremental development, and continuous collaboration between development teams and stakeholders. Unlike traditional models such as the Waterfall model, which follows a rigid, linear process, Agile breaks the project into smaller, manageable iterations called "sprints." Each sprint typically lasts two to four weeks and results in a functional version of the software.

Agile emphasizes adaptive planning, where requirements and solutions evolve based on user feedback. This approach enables development teams to respond quickly to changing customer needs, technological advancements, and market dynamics. Continuous integration and testing are key aspects of Agile, ensuring that software quality is maintained throughout the development process.

### **Core Principles of Agile**

* **Customer Collaboration:** Active involvement of stakeholders to refine requirements and validate features.
* **Iterative Development:** Delivering software in incremental cycles to ensure steady progress.
* **Flexibility:** The ability to adapt to changes in requirements at any stage of development.
* **Continuous Feedback:** Regular interactions with users to improve the product.
* **Cross-functional Teams:** Developers, designers, and testers work together throughout the development lifecycle.

### **Advantages of Agile Model**

* Faster delivery of functional software due to incremental releases.
* High customer satisfaction through regular updates and feedback loops.
* Better risk management since issues can be identified and addressed early.
* Improved adaptability to changing requirements.

### **When to Use Agile**

* Projects with dynamic or evolving requirements.
* Medium to large-scale applications requiring frequent updates.
* User-centric applications where customer feedback is crucial.

### **Why Agile Model was Chosen**

The Agile Model was selected for the "Expense Splitter: Simplifying Group Expense Management" project due to its adaptability, incremental development approach, and focus on continuous improvement. This model aligns perfectly with the project's requirements to create a dynamic, user-centric application with advanced features like multi-currency support, debt management, and social connectivity.

#### **Reasons for Selecting Agile:**

1. **Project Type and Size:**
   * The project is small to medium-sized but complex in terms of user-centric functionalities like payment integration and multi-currency transactions. Agile supports iterative development, making it suitable for such projects.
2. **Flexibility and Risk:**
   * Requirements may change based on user feedback, especially for features like recurring expense tracking and UI enhancements. Agile accommodates these changes without significant disruption.
3. **Stakeholder Involvement:**
   * Continuous collaboration with stakeholders, including users and developers, ensures that the product remains aligned with user needs.
4. **Timeline and Budget:**
   * With a strict timeline of four months to launch the MVP, Agile's sprint-based approach allows the development team to deliver functional features incrementally and efficiently manage resource allocation.

### **Implementation of Agile in the Project**

1. **Sprint Planning:**

* Divide the project into multiple sprints (e.g., four two-week sprints). Each sprint will focus on developing and delivering specific features:
  + - Sprint 1: Core expense management and group creation
    - Sprint 2: Debt management and recurring expenses
    - Sprint 3: Payment gateway integration
    - Sprint 4: Multi-currency support and data export in PDF

1. **Daily Stand-ups:**
   * Short daily meetings to track progress, identify roadblocks, and discuss solutions.
2. **Backlog Management:**
   * Maintain a prioritized backlog of user stories using tools like Trello or Jira.
3. **Feature Prioritization:**
   * Use MoSCoW prioritization to focus on essential features first (Must-have, Should-have, Could-have, Won’t-have).
4. **Incremental Releases:**
   * Deliver functional versions of the app at the end of each sprint for stakeholder testing and feedback.
5. **Continuous Feedback:**
   * Gather user feedback after each sprint and incorporate necessary changes in subsequent iterations.
6. **Retrospectives:**
   * Conduct sprint retrospectives to evaluate what went well, identify challenges, and implement improvements.

This approach ensures the development of a flexible, feature-rich, and user-friendly expense management application, meeting the evolving needs of users while adhering to the project timeline and constraints.

### **Learning Objective:**

* To understand different software development models and their characteristics.
* To analyze the pros and cons of various models.
* To select the most suitable model for the project based on the findings.

### **Learning Outcome:**

At the end of this experiment, students will be able to:

1. Identify and describe various software development models.
2. Evaluate the suitability of each model for specific project scenarios.
3. Select and justify the choice of a model for their project.

### **Course Outcomes (COs):**

* CO1: Understand and explain the fundamentals of software engineering, the software process frameworks, and umbrella activities to manage and improve software development.
* CO2: Analyze and compare traditional and agile software development models, including their applicability to various project scenarios.

### **Cognitive Levels of Attainment as per Bloom’s Taxonomy:**

* **L1 (Remember):** Identify and list various software development models.
* **L2 (Understand): Explain the characteristics and advantages of different models.**
* **L4 (Analyze): Evaluate the strengths and weaknesses of models to select the most suitable one for the project.**

### **Programme Outcome (PO) Attainment:**

* **PO1: Engineering Knowledge: Apply knowledge of engineering fundamentals to understand software development models.**
* **PO2: Problem Analysis: Analyze the characteristics of various models to solve complex problems.**
* **PO5: Engineering Tool Usage: Utilize research tools and resources to perform a detailed literature survey.**
* **PO11: Life-Long Learning: Develop the ability to explore and evaluate emerging software engineering models.**

### **Program Specific Outcome (PSO) Attainment:**

* **PSO1: Apply software engineering knowledge to select sustainable and efficient development models for IT projects.**

### **Result & Discussion:**

* **Result**:
  1. Identified and documented the characteristics of various software development models.
  2. Evaluated each model's applicability to the project scenario.
  3. Selected the most suitable model for the project and justified the choice.
* **Discussion:**  
  The literature survey emphasized the importance of understanding different software development models to ensure their proper alignment with project requirements. The selected model balances flexibility, risk, and development speed.

### **Conclusion:**

A systematic literature survey provides a comprehensive understanding of software development models. Selecting the appropriate model ensures that the project adheres to timelines, meets stakeholder expectations, and mitigates potential risks effectively.