

Assignment 1

Software Process Models and Requirements Engineering

Case Study: CampusKart – Online E-Commerce Platform for College Students

Student Name: Anitha Saraswathi K

USN : NNM24IS030

Semester & Section: 4th Semester, Section A

Department: Information Science & Engineering

Institution Name: NMAM Institute of Technology

Academic Year: 2025–2026

Instructor Name: Dr. Jason Elroy Martis

Course Name: Introduction to Software Engineering

Course Code: IS1103-1

Date of Submission: 12-02-2026

GitHub Repository Link

1. Abstract

This report presents a comparative study of three major Software Development Life Cycle (SDLC) models: Waterfall, Incremental, and Spiral. The selected case study is CampusKart, an online e-commerce platform designed for college students to buy and sell academic materials such as books, lab equipment, calculators, and other second-hand items.

The report analyzes how each process model supports functional and non-functional requirements, handles risk and changes, and manages time and cost constraints. A simplified requirements document is also developed as part of the study. Based on the comparison, the Incremental Model is identified as the most suitable approach for developing CampusKart because it allows gradual development, flexibility in handling changing requirements, and early user feedback.

2. Introduction

Software development requires a structured approach to ensure that the system is reliable, cost-effective, and meets user needs. This structured approach is called the Software Development Life Cycle (SDLC).

Different SDLC models are used depending on the nature of the project. Choosing the correct model is important because it affects requirement clarity, risk management, time estimation, and overall project success.

In this assignment, CampusKart is selected as the case study. It is a web-based platform created for college students to buy and sell academic items within the campus. The report compares Waterfall, Incremental, and Spiral models and studies their suitability for developing this system.

3. System Overview – CampusKart

CampusKart is an online e-commerce system specifically designed for students. The main purpose is to create a simple and secure platform for exchanging academic materials.

Main Features:

- User registration and login

- Product listing by students
- Search and filter products
- Add to cart
- Online payment
- Order tracking
- Wishlist
- Rating and review system
- Admin dashboard

Stakeholders:

- Students (buyers and sellers)
- System administrator
- Payment gateway provider
- Delivery partner

Since the system may expand in the future, selecting a flexible development model is important.

4. Functional and Non-Functional Requirements

4.1 Functional Requirements

Functional requirements describe what the system should do.

Examples:

- The system shall allow students to register and login.
- The system shall allow users to add, edit, and delete products.
- The system shall allow users to search products by category or price.
- The system shall allow users to add items to cart.
- The system shall allow users to make online payments.

- The system shall allow users to track orders.
 - The admin shall manage users and remove inappropriate listings.
-

4.2 Non-Functional Requirements

Non-functional requirements define system quality and performance.

Examples:

- System response time should be less than 3 seconds.
- System should support at least 1000 concurrent users.
- System should ensure data security and encryption.
- System should maintain 99% uptime.
- Website should be mobile responsive.

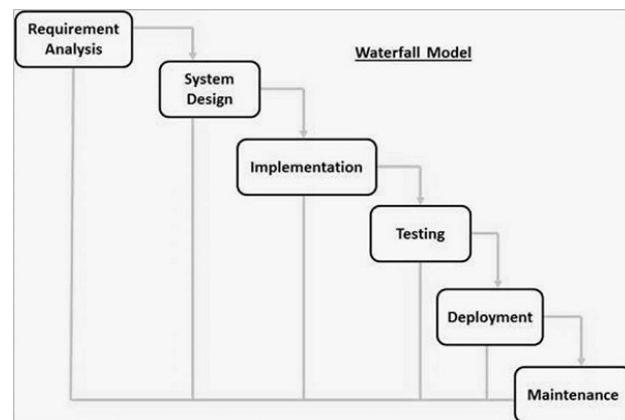
Both functional and non-functional requirements must be clearly defined before development.

5. Comparison of SDLC Models

5.1 Waterfall Model

The Waterfall model is a linear and sequential model. Development flows through different phases in order:

1. Requirement Analysis
2. System Design
3. Implementation
4. Testing
5. Deployment
6. Maintenance



Each phase must be completed before moving to the next.

Suitability for CampusKart

Waterfall works well if requirements are fixed and clearly known at the beginning. If CampusKart only includes basic features and no changes are expected, Waterfall can be used.

However, in real situations, student needs may change frequently.

Impact on Requirements

- Functional requirements must be fully defined at the start.
- Non-functional requirements are difficult to modify later.

Risk and Change Management

- High risk if requirements are misunderstood.
- Changes after design phase are costly.
- Errors are discovered late during testing.

Time and Cost

- Easy to estimate initial cost and timeline.
- Rework increases cost significantly.

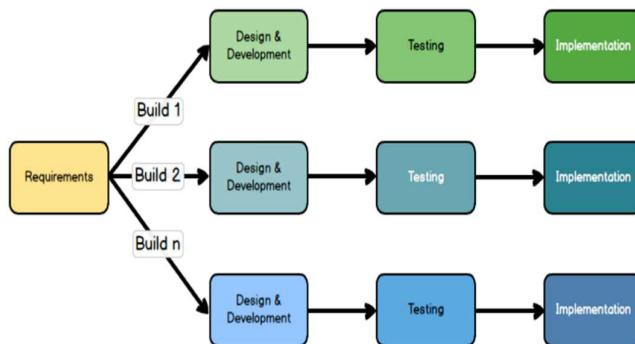
Advantages

- Simple and easy to understand
- Clear documentation
- Suitable for small and stable projects

Disadvantages

- Not flexible
- Late feedback
- Difficult to handle requirement changes

5.2 Incremental Development Model



In the Incremental model, the system is developed in small parts called increments. Each increment includes requirement analysis, design, coding, and testing.

For CampusKart, development can be divided as:

- Increment 1: Registration and login
- Increment 2: Product listing and search
- Increment 3: Cart and payment
- Increment 4: Wishlist and reviews

Each increment delivers a working version of the system.

Suitability for CampusKart

This model is highly suitable because:

- Students can start using the system early.
- Feedback can be collected after each release.
- New features can be added easily.

Impact on Requirements

- Functional requirements can be implemented gradually.
- Non-functional improvements (like performance) can be added later.

Risk and Change Management

- Risk is reduced since testing happens in each increment.

- Requirement changes can be included in future releases.

Time and Cost

- Early working version is available.
- Cost is distributed over multiple phases.
- Easier budget control.

Advantages

- Flexible
- Early delivery
- Better stakeholder involvement

Disadvantages

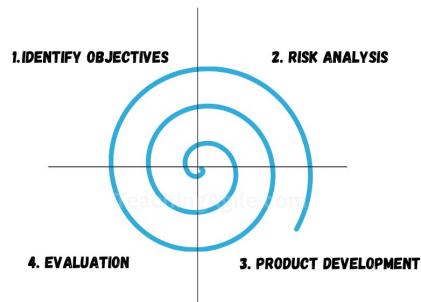
- Requires proper planning
 - Integration issues may arise
-

5.3 Spiral Model

The Spiral model combines iterative development with risk analysis. It focuses mainly on identifying and reducing risks.

Each cycle includes:

1. Planning
2. Risk Analysis
3. Engineering
4. Evaluation



It is suitable for large and complex systems.

Suitability for CampusKart

If CampusKart expands into a large-scale platform with advanced security, multiple integrations, and large user base, Spiral may be useful.

However, for a college-level project, it may be too complex.

Impact on Requirements

- Handles complex functional requirements well.
- Strong focus on non-functional requirements like security and performance.

Risk and Change Management

- Risk is analyzed in every cycle.
- Prototyping reduces uncertainty.

Time and Cost

- Expensive due to continuous risk analysis.
- Time-consuming.

Advantages

- Strong risk control
- Flexible
- Suitable for high-risk projects

Disadvantages

- Complex to manage
- High cost
- Not suitable for small teams

6. Comparative Summary

Criteria	Waterfall	Incremental	Spiral
Requirement Flexibility	Low	Medium	High

Criteria	Waterfall	Incremental	Spiral
Risk Handling	Low	Medium	High
Cost	Low initially	Moderate	High
Time to Market	Late	Early	Moderate
Suitability for CampusKart	Moderate	Highly Suitable	Suitable for large scale

Based on the comparison, the Incremental model is the most suitable for CampusKart.

7. Requirements Engineering Process

Requirements Engineering consists of:

1. Elicitation
 2. Analysis
 3. Documentation
 4. Validation
 5. Management
-

7.1 Requirements Elicitation

Requirements were collected using:

- Student surveys
 - Informal interviews
 - Observation of existing e-commerce platforms
 - Brainstorming sessions
-

7.2 Requirements Analysis

- Removed conflicting requirements

- Categorized into functional and non-functional
 - Prioritized important features
-

7.3 Simplified Requirements Document

Functional Requirements

- User shall register and login.
- User shall list products.
- User shall search and filter products.
- User shall add items to cart.
- User shall make secure online payment.
- User shall track orders.
- Admin shall manage users.

Non-Functional Requirements

- System response time < 3 seconds
 - Support 1000 concurrent users
 - Secure payment processing
 - 99% uptime
 - Mobile responsive design
-

8. Requirements Validation Strategy

Validation ensures that requirements are correct and complete.

Methods used:

- Review meetings
- Prototype demonstration

- Requirement checklist
 - Stakeholder approval
-

Challenges in Validation

- Ambiguous requirements
 - Changing user expectations
 - Communication gaps
 - Ignoring non-functional requirements
 - Limited stakeholder availability
-

9. Conclusion

This report analyzed three SDLC models – Waterfall, Incremental, and Spiral – for the development of CampusKart.

Waterfall is simple but not flexible. Spiral provides strong risk handling but is costly and complex. Incremental model offers flexibility, early delivery, and manageable risk.

Therefore, for CampusKart, the Incremental Development Model is the most practical and suitable choice. Proper requirements engineering and validation play an important role in ensuring successful system development.
