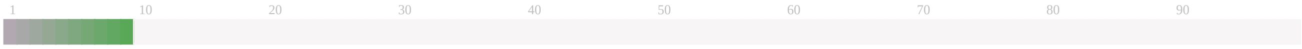
The Report is Generated by DrillBit Plagiarism Detection Software

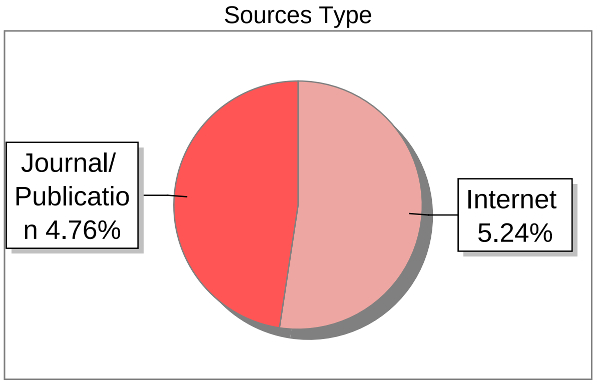
Submission Information

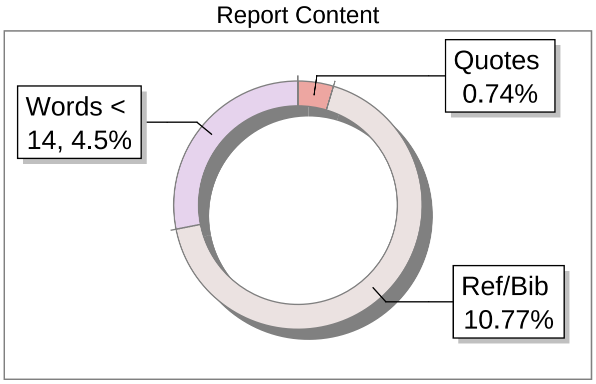
Author Name	Akash Nayak
Title	nnm24is015_task1
Paper/Submission ID	5271775
Submitted by	nnm24is040@nmamit.in
Submission Date	2026-02-15 19:55:35
Total Pages, Total Words	9, 2312
Document type	Assignment

Result Information

Similarity **10 %**




Sources Type


Report Content

Exclude Information

Quotes	Not Excluded
References/Bibliography	Not Excluded
Source: Excluded < 14 Words	Not Excluded
Excluded Source	**0 %**
Excluded Phrases	Not Excluded

Database Selection

Language	English
Student Papers	Yes
Journals & publishers	Yes
Internet or Web	Yes
Institution Repository	Yes

A Unique QR Code use to View/Download/Share Pdf File

DrillBit

DrillBit Similarity Report

10	18	A	A-Satisfactory (0-10%) B-Upgrade (11-40%) C-Poor (41-60%) D-Unacceptable (61-100%)
SIMILARITY %	MATCHED SOURCES	GRADE	

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	businessperspectives.org	1	Internet Data
2	redcol.minciencias.gov.co	1	Publication
3	Thesis Submitted to Shodhganga Repository	1	Publication
4	www.scnsoft.com	1	Internet Data
5	How to steer an embedded software project tactics for selecting the s by Petr-2005	1	Publication
6	baixardoc.com	1	Internet Data
7	documents.mx	<1	Internet Data
8	jurnal.murnisadar.ac.id	<1	Internet Data
9	docplayer.net	<1	Internet Data
10	docplayer.net	<1	Internet Data
11	Locality-Based Network Creation Games by Bil-2016	<1	Publication
12	pdfcookie.com	<1	Internet Data
13	Understanding the atomic and electronic origin of mechanical property, by San, Saro Li, Neng- 2018	<1	Publication

14	www.motionfitnessokotoks.ca	<1	Internet Data
15	arxiv.org	<1	Publication
16	jurnal.ugm.ac.id	<1	Internet Data
17	kuey.net	<1	Publication
18	uktech.ac.in	<1	Publication

Railway Management Reservation System SDLC Comparative Study Protocol

Course name: Introduction to Software Engineering

Course code: IS3332-1

Assignment Number & Title: SDLC Comparative Study – Railway Reservation Management System

Student Name: Akash Narasimha Nayak

USN / Roll Number: NNM24IS015

Semester & Section: IV A

Department: Information Science & Engineering

Institution Name: NMAMIT

Academic Year: 2025–2026

Instructor Name: Jason Elroy martin

Date of Submission: 15/02/2026

2. ABSTRACT

For this assignment, I chose the Railway Reservation Management System as a case study to examine several Software Development Life Cycle (SDLC) models. The primary aim was to determine which development model is best suited for a large-scale, real-time system. I evaluated the Waterfall, Incremental Development, and Spiral models based on their flexibility in requirements, risk management capabilities, and considerations for cost and time constraints. I also created a simplified requirements document and outlined the validation strategy. Based on my analysis, I discovered that the Waterfall model does not provide adaptability for systems with changing requirements. The Incremental and Spiral models are more suitable, especially for systems managing online payments and facing significant traffic. In conclusion, I determined that Incremental development with effective risk management is the most viable option for the Railway Reservation Management System.

2. GITHUB REPOSITORY LINK

url:

https://github.com/AkashNayak1223/SDLC-Railway-System

3. INTRODUCTION

The railway booking system is an online platform that helps the users to find trains, purchase tickets, and secure reservations through the internet.. [1] As the growth of digital technology increases, it affects many fields, including the tourism industry. The application of Information and Communication Technology (ICT) improves the management of tourist destinations, the ticket booking process, and the digital promotional process. [2] The Software Development Life Cycle (SDLC) is a systematic approach to software development that follows a series of steps, which include planning, system design, implementation, requirements analysis, testing, deployment, and maintenance. [3]

The aim of this assignment is to explore and analyze different SDLC models by applying them to a real-world system. The Railway Reservation Management System was chosen because It is a complicated system that needs careful planning, effective risk management, and efficient requirements management. This assignment has helped me understand the effect of different development methodologies on the successful outcome of a project and how changes in requirements can affect the project's cost and schedule.

4. PROBLEM STATEMENT

The Railway Reservation Management System is a software system that is complex and operates in real time. It should be able to handle several operations such as user login, finding trains, checking seats, ticket booking, online payment, ticket cancellation, etc. Developing such a system is difficult because it has many users and requires real-time updates. The system should also be reliable and secure. The Software Development Life Cycle (SDLC) model should be chosen properly to meet the challenges and complexities involved in developing such a system. [4]

Task to be Solved

The task to be solved is to analyze and compare the numerous models such as the Incremental model, a Waterfall model and a Spiral model and determine which one is the most appropriate to enlarge the

Railway Reservation Management System. The analysis would be done based on the flexibility in the requirements, the level of risk, the cost of development, and the time limits. [5]

Expected Outcome

The outcome of the analysis would be the identification of the proper SDLC model that would result in the development of a reliable, secure, and scalable system. [4,6]

Constraints

The system should be developed with the ability to handle many users at the same time, should be updated in real time, and should also be secure enough to handle online payments. The other constraints could be the budget allocated to the developers, the time limits, and the changes in the requirements due to changes in policies and rules. [7]

5. SYSTEM OVERVIEW

The Railway Reservation Management System consists of several interconnected modules that collaborate to oversee the entire reservation procedure.

This system features the following modules:

- Authentication Module
- Train Search Module
- Booking Module
- Payment Module
- Cancellation Module
- Admin Module

The authentication module is utilized for secure access, the train search module assists users in locating available services, the booking module enables seat reservation, the payment module handles online transaction processing, the cancellation module deals with ticket cancellations, and the administrative module oversees train schedules and user records. The system has been created to function reliably under heavy user traffic, scale effectively as demand grows, and maintain robust security to safeguard sensitive user and transaction information.

6. COMPARATIVE ANALYSIS OF SLDC MODELS

6.1 Waterfall Model

The Waterfall model follows a sequential development process, in which each phase needs to be finished prior to starting the subsequent one.[8] Nevertheless, it lacks adaptability when project requirements evolve during the development process. [4] It is effective when the requirements are explicitly established and remain constant at the outset of the software development lifecycle. [8,5] However, in a relation of Railway Reservation Management System, requirements can frequently shift due to updates in policies, advancements in technology, or the introduction of new features, rendering the Waterfall model less suitable. [4]

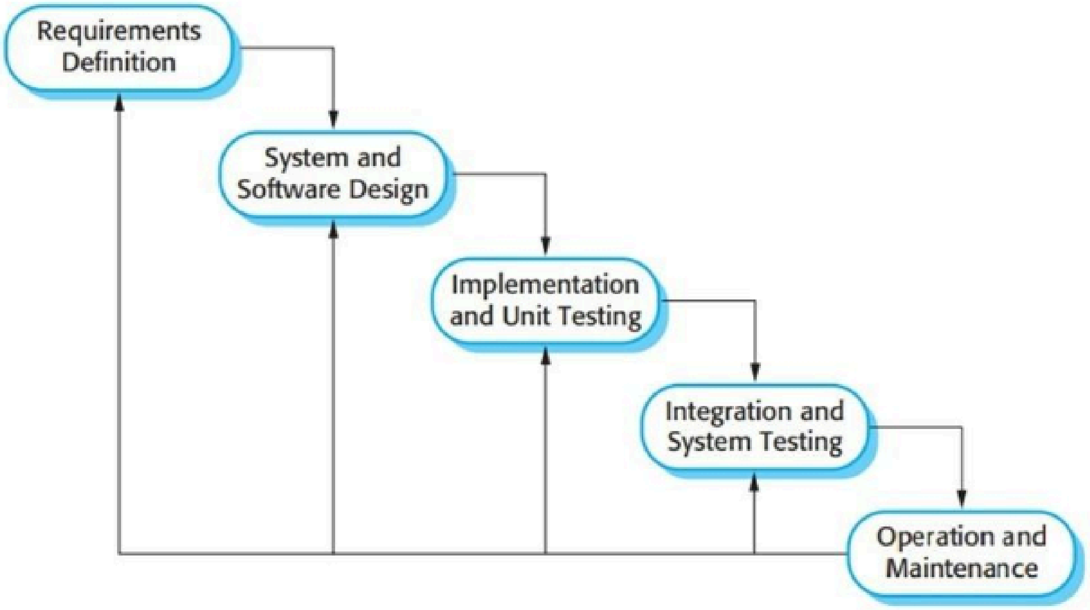

Requirements Definition
System and Software Design
Implementation and Unit Testing
Integration and System Testing
Operation and Maintenance

Figure 1- Waterfall model [4]

Suitability:

The Waterfall model is not ideal for railway reservation systems since it struggles to incorporate changes after the development has commenced. [8,4]

6.2 Incremental Developmental Model

In this model, the system is developed in smaller parts (increments).

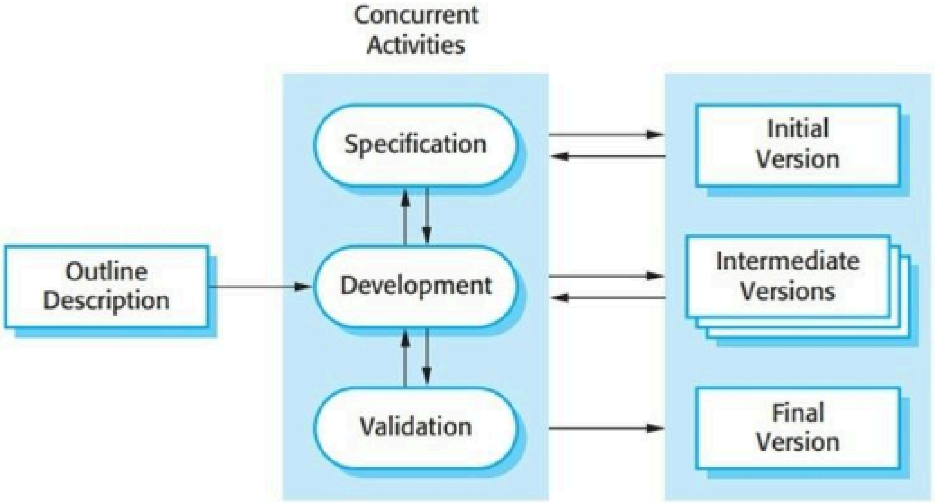


FIGURE 2- Incremental Development Model [4]

Example:

- Login system
- Train search
- Booking and payment
- Cancellation

Each increment is completed and tested before adding the next one.

Suitability:
More suitable because changes can be made in later increments and risk is reduced.

6.3 Spiral Model

The Spiral model employs risk analysis and development cycles to enhance the system with each development iteration.[9] The development cycles include planning, risk analysis, development, and evaluation. These cycles enable the identification of possible problems with the system at a initial phases of development.[9] This model can be very useful in systems with high-risk components, for example, in online payment systems characterized by high traffic volume, which is common in railway reservation systems.[7]

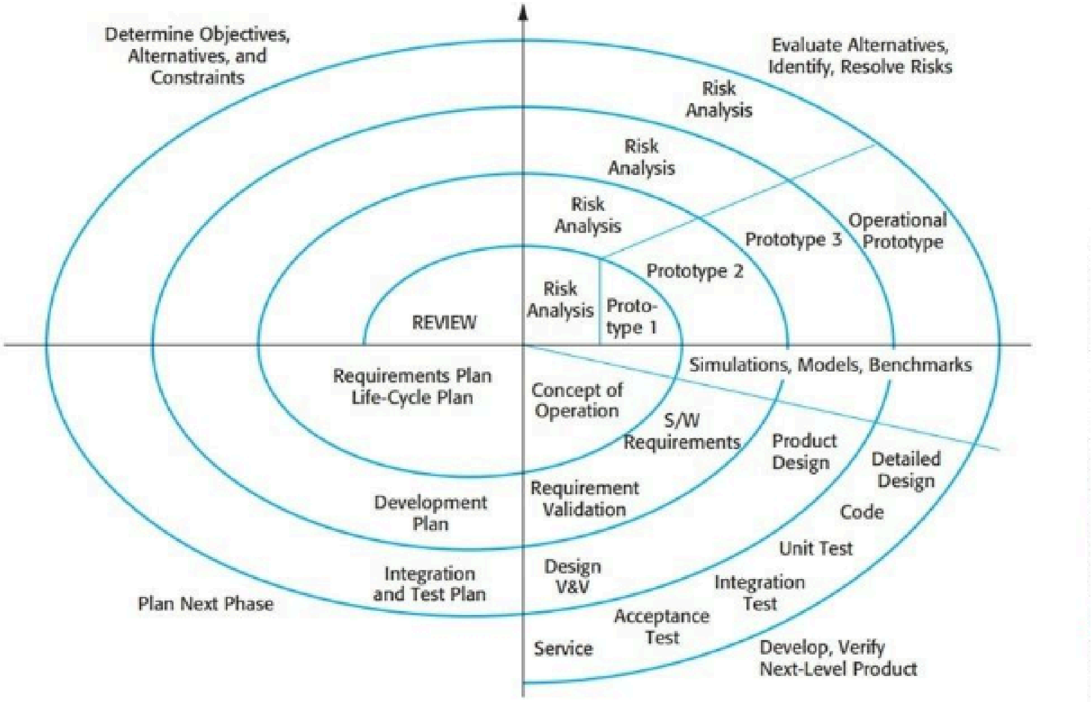


FIGURE 3 – Spiral model [4]

Suitability:

The Spiral model is suitable for large complex systems due to its emphasis on risk management. However, it is more expensive and difficult to implement compared to other SDLC models. [4,9]

7. REQUIREMENTS ENGINEERING:

Requirements engineering is the activity of seeking out, analyzing, documenting, and validating the non-functional and functional requirements of a software system. [6]

7.1 Functional Requirements

Functional requirements outline the actions the system must perform and the services it needs to offer to users. [4]

1. Users must be able to create an account and sign in.
2. Users should be able to look up trains based on their origin and destination.
3. It needs to show the availability of seats.
4. Should facilitate the booking of tickets.
5. System must securely handle online payment processing.
6. Users should have the capability to cancel their tickets.

7. Administrators should oversee train schedules and manage user information.

These functional requirements are common for online railway reservation systems, as highlighted in research on web-based ticketing and reservation services. [10-11]

7.2 Non-functional Requirements

It specifies a status of the system's performance, encompassing constraints related to performance, security, reliability, and availability. [6]

1. The system must accommodate multiple users at the same time.
2. The response time needs to fall within acceptable ranges.
3. The system must guarantee data protection.
4. Regular backups of the database are essential.
5. The system should ensure consistent availability.

Non-functional requirements like performance, security, and availability are essential for large-scale systems such as railway reservation platforms that manage high traffic and financial operations. [5,7]

8. REQUIREMENTS VALIDATION STRATEGY

Validation of requirements verifies that the requirements are accurate, complete, and unambiguous, and that they are agreed upon by all stakeholders. [4] Requirement review meetings are held to ensure that there are no errors, discrepancies, or missing information. [6] Verifying requirements for consistency and completeness ensures that all functional and non-functional requirements are well defined and do not conflict with one another. [5]

Gaining the approval of stakeholders is an important part of requirements validation, which verifies that the requirements satisfy both the users and the organization. [6] Developing simple prototypes of the system can also help stakeholders visualize the behavior of the system and understand requirements better. [4] Tracing requirements through design, implementation, and testing makes sure that all requirements are met. [5]

However, requirements validation can also be affected by problems such as ambiguous requirements and changing requirements, which are common in this process. [4]

9. DISCUSSION

From this study, I learned that choosing the right SDLC model depends on the system type and its complexity. The Railway Reservation Management System faces common requirement changes and security risks. Therefore, the Incremental and Spiral models are better options than Waterfall. The Incremental model is more practical because it balances flexibility with cost.

The Railway Reservation Management System is complex and operates in real-time. It must handle many users, continuous transactions, and frequent operational updates. Because of these traits, picking the right Software Development Life Cycle (SDLC) model is crucial. It affects system reliability, flexibility, and maintainability. I analyzed different SDLC models to see which ones fit best for developing such a large-scale system. [4]

The Waterfall model uses a linear and sequential development process. Each round should be finished before the beginning the next one. This model works well when system requirements are clear and stable.

[8] However, in a railway reservation system, requirements often shift due to policy updates, security enhancements, fare changes, and user demands. Since the waterfall model do not easily accommodate changes once development is underway, it is less effective for dynamic systems like railway reservation platforms. [5]

The Spiral model addresses some of these limitations. It merges iterative progress with thorough risk assessment. Each cycle includes planning, risk assessment, development, and evaluation. This helps recognize and address potential risks early in the process. [9] This approach is particularly useful for railway reservation systems, where online payment risks, data security issues, and high user traffic are significant. However, the Spiral model needs extensive planning, expert risk assessment, and higher development costs, making it more appropriate for large, high-risk projects rather than cost-sensitive situations. [7]

The Incremental Development Model offers a balanced and practical solution for the Railway Reservation Management System. In this method, the system is developed and delivered in smaller functional parts. This allows for early implementation of key features, such as user registration, train search, and booking. Additional functions, like payment processing, ticket cancellation, and administrative management, can be added gradually. This model supports continuous testing and user feedback, which improves software quality and makes it easier to implement requirement changes. [12] Consequently, incremental development suits systems with changing requirements and long operational lifecycles.

Requirements engineering and validation are also vital for the system's success. Clearly defined functional and non-functional requirements ensure that the system meets user expectations for performance, security, and availability. Validation techniques, like requirement reviews, stakeholder approval, prototyping, and traceability, help reduce ambiguity and ensure that the final system meets user needs. [6] These practices are crucial for minimizing errors and enhancing system reliability.

In conclusion, this discussion shows that flexibility, modular development, and risk management practices are crucial for building a reliable Railway Reservation Management System. Among the SDLC models examined, the Incremental Development Model stands out as the most suitable approach. It supports evolving requirements, improves system quality, and ensures scalability and maintainability throughout the system’s lifecycle. [4]

10. CONCLUSION

In this study, we looked at different Software Development Life Cycle (SDLC) models using the Railway Reservation Management System as a case study. The comparison helped us understand how different development approaches impact system flexibility, risk management, cost, and overall project success.

The Waterfall model is easy to understand but inflexible, making it unsuitable for systems where requirements frequently change. The Spiral model provides strong risk management and works well for complex, high-risk components, but it raises development costs and complexity. The Incremental Development Model offers a balanced approach by allowing the system to be built in smaller modules with ongoing testing and feedback.

Given the size, complexity, and changing requirements of a railway reservation system, the Incremental model stands out as the most practical option. It supports gradual development, lowers risk, and makes it easier to incorporate changes while ensuring system reliability and quality.

REFERENCES:

1. -, S.M., -, S.S., -, S.T., & -, V.S. (2023). Railway Reservation System using PHP. *International Journal For Multidisciplinary Research*.
2. D. Zatnika and N. Rukhviyanti, "Application of Forward Chaining Method in Systems Expert Recommends Second Cars from the Aspect of Job Income," Research Journal Inovatif, vol. 4, no. 4, pp. 2463–2476, Dec. 2024, doi: 10.54082/jupin.759.
3. Pressman, R. S. (2005). *Software engineering: a practitioner's approach*. Palgrave macmillan.
4. Sommerville, I. (2011). *Software engineering, 9/E*. Pearson Education India.
5. Pressman, R., Maxim, B., & Fowler, M. (2025). Software Engineering: A Practitioner's Approach.
6. Wiegers, K., & Beatty, J. (2013). *Software requirements*. Pearson Education. [12]
7. Boehm, B. W. (1991). Software risk management: principles and practices. *IEEE Software*, *8*(1), 32-41.
8. Royce, W. W. (1970). Managing the development of large software systems. proceedings of IEEE WESCON, 1970. *Los Angeles*, 1-9.
9. Boehm, B. W. (1988). A spiral model of software development and enhancement. *Computer*, *21*(5), 61-72.
10. Zongjiang, W. (2012). Railway online booking system design and implementation. *Physics Procedia*, *33*, 1217-1223.
11. Kumar, R. B., & Singh, A. (2023). Design and implementation of online railway ticket reservation system. *International Journal of Engineering and Advanced Technology*, 12(3), 224–229.
12. Larman, C., & Basili, V. R. (2003). Iterative and incremental developments. a brief history. *Computer*, *36*(6), 47-56.

DECLARATION OF ACADEMIC INTEGRITY

I hereby declare that this report is my original work. All references used have been properly cited. No part of this work has been copied without acknowledgment.

Signature:

Date: 15-02-2026