## Submission Information

| Author Name | Deepthi N Raikar |
|---|---|
| Title | Task |
| Paper/Submission ID | 3335285 |
| Submitted by | nnm23is030@nmamit.in |
| Submission Date | 2025-02-16 08:11:56 |
| Total Pages, Total Words | 14, 2816 |
| Document type | Project Work |

## Result Information

Similarity **6 %**



### Sources Type

Student Paper 0.32%
Journal/Publication 1.38%
Internet 4.3%

### Report Content

Ref/Bib 0.99%
Words < 14, 3.59%
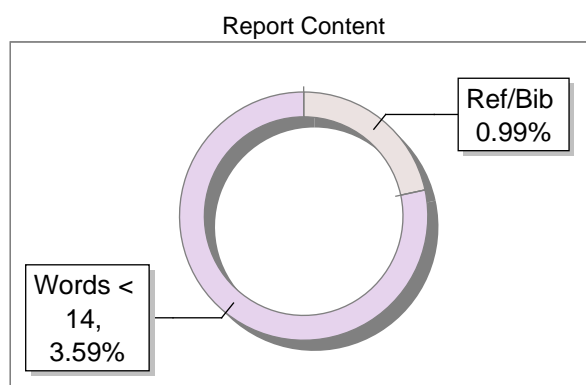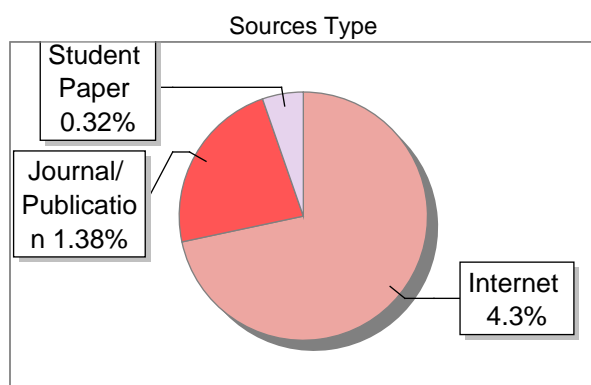
## Exclude Information

| | | | |
|---|---|---|---|
| Quotes | Not Excluded | Language | English |
| References/Bibliography | Not Excluded | Student Papers | Yes |
| Source: Excluded < 14 Words | Not Excluded | Journals & publishers | Yes |
| Excluded Source | **0 %** | Internet or Web | Yes |
| Excluded Phrases | Not Excluded | Institution Repository | Yes |

## Database Selection

# DrillBit

| | | | |
|:---:|:---:|:---:|:---|
| **6** | **12** | **A** | **A-Satisfactory (0-10%)** |
| SIMILARITY % | MATCHED SOURCES | GRADE | **B-Upgrade (11-40%)** |
| | | | **C-Poor (41-60%)** |
| | | | **D-Unacceptable (61-100%)** |

| LOCATION | MATCHED DOMAIN | % | SOURCE TYPE |
|:---|:---|:---:|:---|
| **1** | testsigma.com | 1 | Internet Data |
| **2** | resumeworded.com | 1 | Internet Data |
| **3** | data.unicef.org | 1 | Publication |
| **4** | www.comfygen.com | 1 | Internet Data |
| **5** | files.eric.ed.gov | <1 | Publication |
| **6** | e-journal.unmas.ac.id | <1 | Internet Data |
| **7** | epdf.pub | <1 | Internet Data |
| **8** | fdokumen.id | <1 | Internet Data |
| **9** | pdf4pro.com | <1 | Internet Data |
| **10** | REPOSITORY - Submitted to Exam section VTU on 2024-07-31 16-10 911885 | <1 | Student Paper |
| **11** | www.arxiv.org | <1 | Publication |
| **12** | ymj.kr | <1 | Internet Data |

# Software Development Lifecycle (SDLC) Analysis of Myntra

A Comparative Study of Different Models in Relation to Myntra's Software Development

https://github.com/deepthiraikar/ISE-Task-1

Deepthi N Raikar

Nitte Mahalinga Adyantaya Memorial Institute of Technology

{nnm23is042@nmamit.in}

| | |
|---|---|
| **Keywords:** | **SDLC, Myntra, Integration, Testing, Scalability, Requirements validation** |

| | |
|---|---|
| **Abstract:** | This research compares the Waterfall, Incremental, and Spiral Software Development Life Cycle (SDLC) models to see which is best suited for Myntra, a well-known fashion e-commerce site. The report focuses on each model's benefits, drawbacks, and implementation techniques in context with Myntra's large and dynamic operations to identify the ideal plan of approach. Despite being well-organized and documented, the Waterfall Model is not flexible enough to adapt to a changing e-commerce platform. Iterative improvements and quicker feature releases are made possible by the incremental model, but careful integration planning is necessary. With its risk-driven iterative methodology, the Spiral Model is found to be the most appropriate because of its capacity to handle inconsistencies and changing requirements. The report also highlights the importance of strong requirements engineering and validation techniques to guarantee scalability, security, and an ideal user experience. The results justify the use of the Spiral Model in Myntra's software development, guaranteeing flexibility, risk control, and sustained performance. |

| | |
|---|---|
| **Publishing:** | This paper is hosted on a GitHub repository, along with the material used for preparing this research. |

# Contents

# 1. Introduction

With its web site and mobile apps, Myntra, one of India's top fashion e-commerce sites, offers a smooth buying experience. Selecting the right Software Development Life Cycle (SDLC) model is crucial for ensuring effective development, scalability, and adaptability given the dynamic nature of e-commerce. Choosing an SDLC model is vital since the platform needs constant updates, security improvements, and performance optimizations. To identify whether the Waterfall, Incremental, and Spiral SDLC models are appropriate for Myntra's system development, this research compares them.

## Case Study

One of the top online fashion and lifestyle retailers in India, Myntra provides a large selection of apparel, accessories, and shoes. Millions of people are offered services regularly by the company's extensive e-commerce platform. The platform used by Myntra must manage large product listings, safe transactions, and customized user experiences while maintaining system security and scalability. The case study analyzes which SDLC models are the best fit for Myntra's platform and focuses on the difficulties the company encounters in software development.

# 2. Overview of Myntra

## 2.1 System Overview

The system at Myntra is built to manage an extensive product library, a high volume of daily transactions, and customized user experiences. Key components of the system consist of:

- **User authentication and account management** include managing user profiles, multi-factor authentication, and secure login.
- **Product Catalog:** A flexible collection of clothes, accessories, and footwear that can be sorted and filtered.

- **Order processing:** complete order management, including adding items to the cart, confirming payments, and monitoring deliveries.
- **Integration of Payment Gateways:** Safe handling of transactions using various payment options.
- **Recommendation Engine:** AI-powered recommendations derived on past browsing and purchasing patterns.
- **Customer support system:** portals for self-service, virtual assistants, and live telecommunications providers for enquiries and assistance.
- **Marketing & Promotions**: Combining discounts, deals, and customized advertising efforts.

Choosing the appropriate SDLC model is essential to ensuring effective development and maintainability given the complexity of this system.

## 2.2 Technologies Used

The technological stack used by Myntra has been meticulously chosen to guarantee performance, security, and scalability.

i. **Front-end technologies:** Angular and React.js are used to create a dynamic and responsive user interface.
ii. **Backend Technologies**: Effective server-side processing and API interactions are made possible using Node.js and Python (Django, Flask).
iii. **Database management:** MongoDB (NoSQL) and MySQL (relational) enable the preservation of both structured and unstructured data.
iv. **Cloud services:** Effective hosting services along with high availability are provided by AWS and Google Cloud.
v. **AI/ML Integration:** Predictive analytics and personalized recommendations are accomplished through the implementation of TensorFlow and Scikit-learn.
vi. **Payment Gateway Integration**: Secure transactions are provided by Stripe, PayU, and Razorpay.

All these technologies work together to improve Myntra's security, user engagement, and performance.

# 3. Comparative Analysis of SDLC Models

## 3.1 Waterfall Model for Myntra

The Waterfall Model follows a linear and sequential approach.
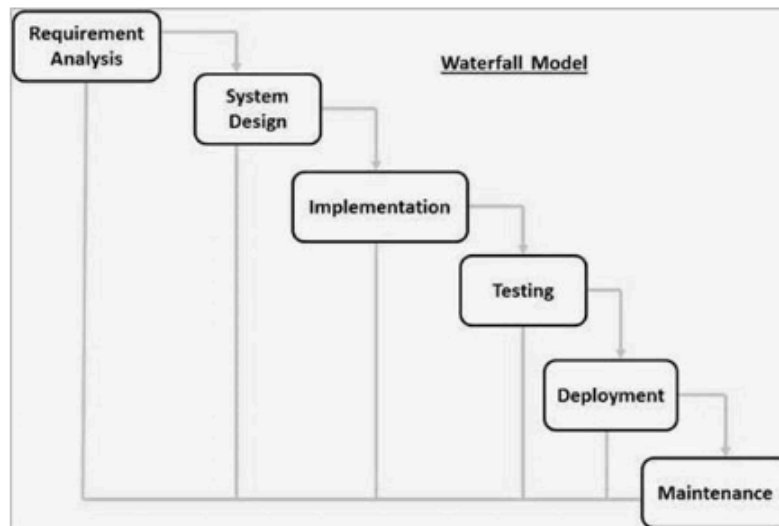


**How Myntra Would Be Developed Using Waterfall**

i.   **Requirement analysis** is the process of collecting and recording all functional and non-functional requirements from technical teams, customers, and business owners, among other stakeholders. For directing growth and development, precise requirement specifications are made.

ii.  **Design**: Database schemas, UI/UX prototypes, and system architecture are developed based on the requirements collected. The selection of technologies and frameworks is based on the scalability requirements and system goals.

iii. **Development**: In the development phase, features are implemented and coded in accordance with the design specification. Best practices are adhered to by developers to guarantee speed optimization, security, and modularity.

iv.  **Testing**: Extensive testing is carried out, comprising user approval, system, integration, and unit testing. Both manual and automated testing make sure the system satisfies its necessary quality requirements.

v.   **Deployment:** To make sure all features work properly, the system is put into a production setting. To manage real-time traffic, security checks and performance monitoring are put in place.

vi. **Maintenance & Continuous Improvement:** The system receives routine updates, security fixes, and performance enhancements after deployment. Feedback from users is gathered to improve functionality and address any problems that may arise.

**Pros**

- ➢ **Clear documentation:** Assures that every stage is thoroughly recorded, which facilitates onboarding and maintenance in the future.
- ➢ **Defined structure**: Uses a step-by-step methodology to eliminate uncertainty and guarantee development clarity.
- ➢ **Ideal for well specified needs:** Reduces scope modifications later and functions best when project criteria are set.

**Cons**

- ➢ **Lack of adaptability:** It is challenging to implement changes once a phase is over, which reduces its flexibility.
- ➢ **Issues are identified late**: Testing takes place after development, which could result in expensive and time-consuming problem solutions.
- ➢ **Inaccurate initial requirements increase the possibility of project failure:** Early requirements that are not recognized make alterations later challenging and increase the probability of failure. Nonetheless, the Waterfall Model guarantees detailed documentation for simpler onboarding and maintenance.

**Final Verdict**

The Waterfall Model is inadequate for Myntra's dynamic e-commerce business due to its strict, sequential methodology. Its lack of adaptability makes it challenging to handle rapid updates, minor modifications, and changing requirements.

## 3.2 Incremental Development Model for Myntra

The Incremental Model involves developing the system in multiple increments.

## Incremental Development Model
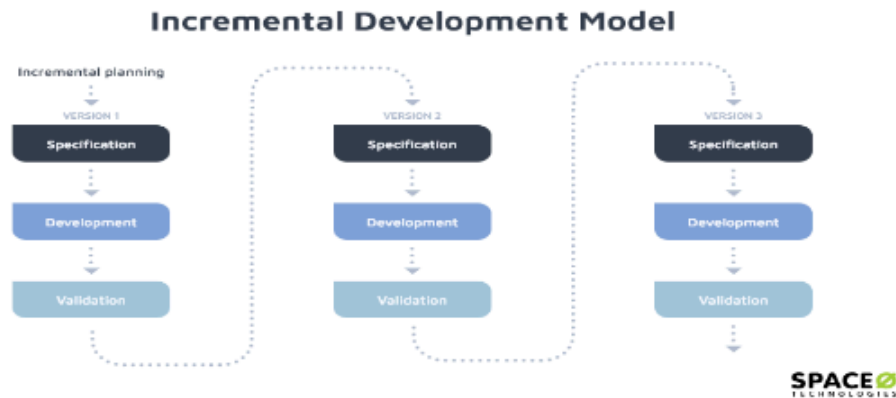


**How Myntra Would Be Developed Using Incremental Development**

i. **Create Core Features:** During the initial stage, the main features that are necessary for Myntra's operation are implemented. This comprises user authentication systems that allow users to safely register and log in, a product catalog that allows users to browse and search for items, and a simple checkout procedure that incorporates secure payment processing.

ii. **Add Advanced Features:** After the fundamental system is solid, more features are added one at a time. AI-powered suggestions are integrated to tailor user experiences according to browsing and buying patterns. Customers may keep an eye on their purchases in real time thanks to the implementation of order tracking tools. Additionally, return management was created to expedite the exchange and refund procedures.

iii. **Enhance the System Based on User Feedback:** To stay competitive, the system must be updated continuously. To find errors and improve usability, user feedback is studied. Regular implementation of security updates, feature enhancements, and performance optimizations improves the user experience overall and guarantees the platform's continued scalability and efficiency.

**Pros**

➢ **Faster time-to-market:** Provides continuous upgrades and faster feature releases.
➢ **Lower failure risk:** Problems are identified early, which reduces project risks overall.

7

- ➤ **Simpler debugging and problem solving:** Troubleshooting is easier with smaller, incremental updates.

**Cons**

- ➤ **Requires meticulous planning**: roadmaps that are well-structured are necessary for incremental releases.
- ➤ **Complexities of integration**: It can be difficult to combine new features with pre-existing ones.
- ➤ **Could increase project costs overall:** Several iterations could require more testing and resources.

**Final Verdict**

With its continuous feature releases and quicker time-to-market, the incremental model is a good choice. By detecting challenges early, it lowers the risks and guarantees ongoing improvement. Cost factors and integration challenges, however, need to be properly handled.

## 3.3 Spiral Model for Myntra
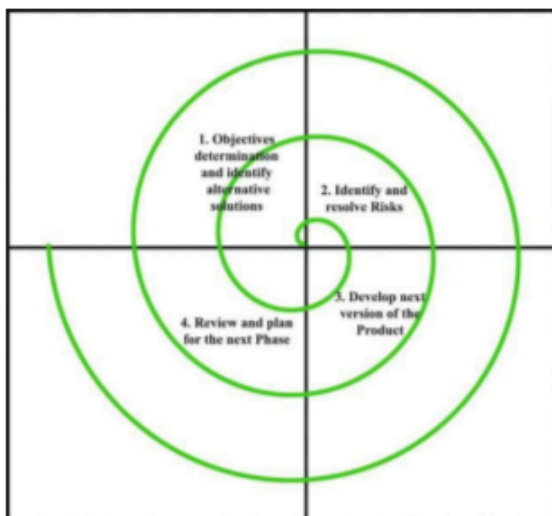


The Spiral Model combines iterative development with risk analysis.

**How Myntra Would Be Developed Using Spiral**

1. **Planning:** In this phase, all system requirements—both functional and non-functional—are determined. Stakeholders including developers, business analysts, and end users work together to establish precise project objectives and uncover any potential risks that can affect progress.
2. **Risk analysis:** This stage examines technological sustainability, scalability issues, and security vulnerabilities. Prevention measures are developed when potential risks including financial restrictions, performance constraints, and cyber threats are examined.
3. **Development of the Prototype:** An initial prototype of the system is built with the focus on its key features and user interface. By enabling stakeholders to see how the system will look and work, this prototype makes sure that the requirements match the objectives of the company.
4. **Evaluation:** The prototype is put through a thorough testing process and user input is gathered. Stakeholders evaluate its security, usability, and performance, providing suggestions for any necessary improvements. To guarantee ongoing improvement, requirements are revised considering this assessment.
5. **Iterative Refinement**: The cycle is repeated, each time adding new features and improvements. Before the system is finished, extensively tested, and prepared for deployment, it goes through several iterations.

**Pros**

- ➤ **Effective risk management:** Identifies and mitigates potential risks early in development.
- ➤ **Iterative improvements:** Enhances the system gradually with user feedback and testing.
- ➤ **Suitable for large, complex projects:** Best for projects with evolving requirements and high complexity.

**Cons**

- ➤ Early detection and mitigation of potential risks is an essential component of **effective risk management.**
- ➤ **Iterative improvements**: Using testing and user feedback, the system is progressively improved.
- ➤ **Ideal for large, complicated tasks**: Best for projects with high complexity and changing requirements.

**Final Verdict**

The Spiral Model is the best choice for Myntra. Because it is continual, it may be continuously improved upon in response to user input and risk assessments. This is essential for managing security issues, improving performance, and adapting to changing business requirements. For a large-scale e-commerce platform, the benefits surpass the drawbacks even though it requires a larger budget and skilled risk analysts.

## 3.4 Summary of Comparison

| Model | Pros | Cons |
|---|---|---|
| Waterfall | Structured, clear documentation | Rigid, costly to modify |
| Incremental | Faster releases, flexible | Higher integration complexity |
| Spiral | Risk-focused, iterative | Expensive, requires expertise |

# 4. Requirements Engineering

Requirements engineering is the systematic process of collecting, analyzing, verifying, and documenting software requirements. It ensures that the finished system satisfies user expectations and business requirements. Because it includes managing large-scale transactions, including AI-driven recommendations, and providing security, this procedure is essential to Myntra's software development. Functional and non-functional requirements, validation techniques, and problem-solving techniques are all part of the requirements engineering process, which guarantees a dependable and expandable e-commerce platform.

## 4.1 Functional Requirements

The functional requirements listed in the report describe the key features that Myntra's e-commerce platform must support to ensure smooth and efficient operations. These requirements cover several kinds of platform features, including transaction processing, product discovery, user management, and post-purchase services.

i. **User authentication and registration** make it possible to create, log in, and control access to user accounts effectively.

ii. **Product Catalog Browsing and Filtering**: This feature enables users to look for and filter products according to several factors, such as brand, price, and category.

iii. **Shopping Cart and Checkout:** Makes it easier to choose products, place orders, and check out quickly.

iv. **Secure Payment Processing**: This guarantees that transactions are safeguarded using encryption and a variety of payment options.

v. **AI-Powered Customized Suggestions:** AI is used to make product recommendations based on user browsing history and preferences.

vi. **Order tracking and return management:** Offers a simple return/refund procedure together with real-time order status notifications.

vii. **Customer service using chatbots and live agents**: These tools help consumers with concerns and queries by providing both automated and human assistance.

viii. **Automated Restocking Alerts and Inventory Management**: These features assist retailers in monitoring stock levels and getting notifications when restocking is required.

ix. **Promotional Campaign Management for Sales and Offers:** This tool helps companies to design and oversee marketing campaigns and discounts.

x. **Vendor and Seller Management for Marketplace Operations**: provides assistance with placing orders and managing accounts to third-party organizations.

These requirements guarantee that the system maintains operational effectiveness while satisfying user expectations.

## 4.2 Non-functional Requirements

i. **Scalability**: Using load distribution and auto-scaling, the system must effectively manage high traffic during seasons of peak sales.
ii. **99.9% Uptime**: This guarantees constant availability through real-time monitoring and backup procedures.
iii. **Secure Transactions**: Uses multi-factor authentication, SSL, and encryption to protect data.
iv. **Quick Response Time:** Performance optimization guarantees that page loads and interactions take place in less than two seconds.
v. **Cross-Platform Compatibility**: Guarantees a smooth experience on iOS, Android, and the web.
vi. **Regulatory Compliance**: Protects user privacy by adhering to GDPR and other data protection rules.

These specifications improve the user experience, security, and dependability of Myntra.

## 4.3 Requirements Validation Strategy

By using a requirements validation strategy, Myntra can make sure that its software development process complies with technical specifications, user expectations, and business demands. Below is a synopsis of each validation strategy:

i. **Stakeholder interviews** are used to get feedback and improve requirements prior to implementation by interacting with customers, developers, and business owners.
ii. **Prototyping:** The process of creating early operating models or UI mockups to illustrate features and gather input for enhancements.
iii. **Using case analysis** focuses on actual user interactions to make sure that all predicted workflows are addressed and that there are less functional gaps as possible.
iv. **Automated testing** reduces human error and ensures consistency by using scripts and testing tools to verify essential features.
v. **Security audits**: Evaluate data security protocols to find flaws and guarantee conformity to security guidelines.
vi. **Performance testing:** Evaluates the system's stability, speed, and responsiveness under different circumstances in order maintain its best performance.

All of these validation methods work together to minimize development risks, detect possible problems early, and guarantee an error-free user experience.

## 4.4 Challenges in Requirements Validation

   i.  **Modifying Business Requirements That Demand for Regular Updates**
       Consumer preferences and industry trends change quickly in a dynamic e-commerce setting like Myntra. To support new features, product categories, and marketing tactics, the system must be continuously modified.

  ii.  **High User Expectations for a Smooth Experience and Personalization**
       Consumers demand effortless shopping, fast website loads, and personalized recommendations. Accurate data collection, AI-driven modifications, and optimal system performance are necessary to meet these expectations.

 iii.  **Integration with External Services like Payment Gateways and Third-Party Transportation**
       Numerous external services, including payment processors (Razorpay, Stripe) and transport companies (Delhivery, Blue Dart), must be integrated with the platform. It can be challenging to guarantee smooth communication between these services while preserving system stability.

  iv.  **Maintaining Security Compliance Despite Changing Cyberthreats**
       Myntra must stick to security requirements (such as PCI DSS for payment security) and utilize multi-layer encryption to safeguard user data because of the increase in online fraud and cyberthreats.

   v.  **Effective Management of Large-Scale Data Processing**
       Every day, the platform generates tremendous amounts of data due to its millions of users, transactions, and product listings. Strong infrastructure and scalable database solutions are necessary for processing this data effectively for analytics, managing stocks, and personalization.

## 5. Conclusion

The analysis indicates that the Spiral Model is best suited for Myntra because it provides iterative risk management, enabling ongoing improvement and adjustment to changing business requirements. This model reduces project uncertainty and improves overall system reliability by ensuring that any risks are identified and reduced early. Another excellent option is the incremental development model, which allows for iterative enhancements based on user feedback and stumbled feature releases that offer a faster time to market. However, despite its structure and thorough documentation, the Waterfall Model is firm which makes it less appropriate for an e-commerce platform that needs to be updated and improved quickly. Through the implementation of a clear requirements engineering approach and strong validation techniques, Myntra can guarantee that its software system satisfies user expectations, aligns with the company's objectives, and maintains high security and performance standards.

## 6. References

1. Sommerville, I. (2015). Software Engineering. Pearson.
2. Pressman, R. S. (2019). Software Engineering: A Practitioner's Approach. McGraw-Hill.
3. IEEE Standard for Software Life Cycle Processes. (2020).