**Software Development Lifecycle (SDLC) Analysis of Myntra**

***A comparative study of different models in relation to Myntra’s software development***

**Daniel Leonard**

Nitte Mahalinga Adyantaya Memorial Institute of Technology, Nitte, Karnataka, India

{danielleo1249@gmail.com, [nnm23is039@nmamit.in](mailto:nnm23is039@nmamit.in)}

**https://github.com/**[**dannyleo-05/Software-Development-Lifecycle-SDLC-Analysis-of-Myntra**](https://github.com/dannyleo-05/Software-Development-Lifecycle-SDLC-Analysis-of-Myntra)

Keywords: SDLC, Myntra, E-commerce, Integration, Testing, Scalability, Requirements Validation.

Abstract: Software development needs to be done with structured methodologies to make it scalable, secure, and efficient. This report discusses Software Development Life Cycle (SDLC) models that can be used for Myntra, India's top fashion e-commerce site. It contrasts various SDLC methodologies, talks about requirements engineering, and analyzes challenges encountered by Myntra in software deployment and feature rollouts.

Success for Myntra relies upon its capacity for quick incorporation of new features, customer experience maximization, and scaling during busy shopping periods. Considering the extremely competitive nature of the e-commerce business, the SDLC model chosen should facilitate agility, efficiency, and ongoing innovation. This document also explores functional and non-functional requirements essential for Myntra's expansion, such as security, scalability, and adherence to industry standards. We also touch upon how incremental development and iterative risk analysis optimize Myntra's software growth. By embracing best SDLC practices, Myntra can provide an effortless, enjoyable, and safe shopping experience for its users and continue to hold the market position in India's online fashion segment.

The report also considers the contribution of DevOps to simplify deployment and maintain system reliability. With increasing expansion for Myntra, efficient software development methods will become instrumental in ensuring continued growth and fulfillment of changing customer needs. The report findings will be used as a guideline for software developers and decision-makers intending to implement robust SDLC applications on e-commerce websites.

This report gives insight relevant to software engineers, architects, and researchers involved with the study of SDLC in e-commerce applications.

Publishing: This article was submitted to Dr. Jason Elroy Martis, Associate Professor, Department of Information Science and Technology, NMAM Institute of Technology. Nitte Karnataka, India.This article is also hosted at a GitHub repository, along with the material utilized for preparing this research. The URL to the GitHub Repository is provided in the endnote.

# Table of Contents

1. **Preface**
2. **Introduction**
3. **Overview of Netflix**
   * System Overview
   * Technologies Used
4. **Comparative Analysis of SDLC Models**
   * Waterfall Model
   * Incremental Development Model
   * Spiral Model
   * Summary of Comparison
5. **Requirements Engineering for Netflix**
   * Functional Requirements
   * Non-Functional Requirements
   * Requirements Validation Strategy
   * Challenges in Requirements Validation
6. **Conclusion**
7. **References**

**1. Introduction**

Myntra is one of India's leading e-commerce sites, dealing in fashion, footwear, and lifestyle items. It was established in 2007 and has revolutionized the experience of online shopping by providing an extensive range of products from top domestic and global brands. With a special emphasis on user personalization and interaction, Myntra has incorporated AI-powered recommendations, quick payment methods, and customer-convenient return policies, which makes it a shopper's delight in the country. As a unit of Flipkart, Myntra has reached further and become even more impactful in the fierce e-commerce battleground.

To support seamless services and high-performance scalability, Myntra runs on a high-scale cloud-based infrastructure primarily using Amazon Web Services (AWS). This facilitates the platform to support millions of users online at a time, especially during peak sales seasons like the "End of Reason Sale" and festive season offers. Real-time updates, dynamic inventory control, and enhanced security are facilitated through cloud-based infrastructure, giving a seamless and hassle-free experience to shop. With the intricacy of its platform, Myntra keeps investing in technology innovation to improve user experience and retain its leadership in the industry.

Being a mass-scale e-commerce platform, Myntra needs a clearly defined Software Development Life Cycle (SDLC) to facilitate regular updates, feature development, and system stability. The selection of the appropriate SDLC model is imperative to manage development processes effectively while ensuring quality and customer satisfaction. Alternative SDLC models like the Incremental Development Model, Spiral Model, and Waterfall Model present differential benefits and setbacks, so determining their feasibility considering Myntra's changing business requirements becomes highly critical.

Throughout this report, a comparative discussion of the afore-mentioned SDLC models with respect to assessing their usability in Myntra's software development is carried out. Their impact on shaping the functionality and efficiency of the platform is also reviewed. Further, the report identifies some major issues faced by the development process, such as scalability issues, quick feature rollouts, and system integration issues. By learning about these areas, Myntra can streamline its software development strategy to maintain growth, reduce inefficiency, and provide a hassle-free shopping experience to its customers.

**2. Overview of Myntra**

**2.1 System Overview**

Myntra is an online fashion and lifestyle products marketplace that offers diverse clothing, shoes, accessories, and more. The platform offers support for various devices, individualized recommendations, and a frictionless shopping experience with virtual try-ons and convenient returns. Myntra is an Indian online shopping platform offering a mobile-first shopping experience, with a very high emphasis on user experience and personalization.

**2.2 Technologies Used**

• Cloud Platform: AWS (Amazon Web Services)

• Architecture: Microservices-based

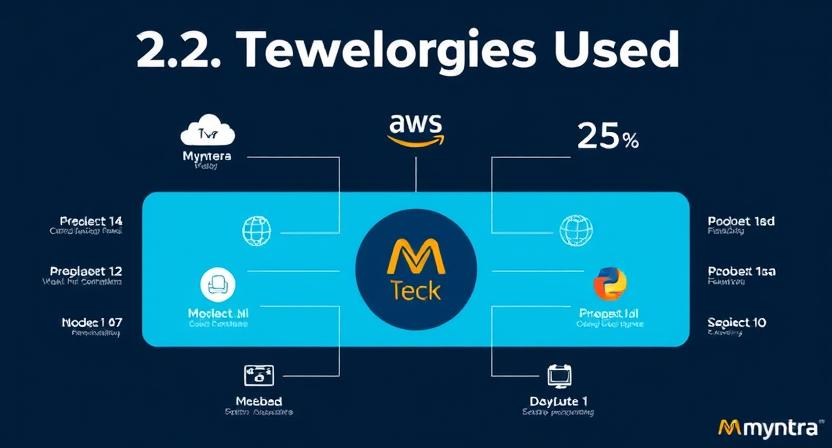
• Database: NoSQL (DynamoDB, Cassandra), MySQL

• Content Delivery: AWS CloudFront, Akamai

• Programming Languages: Java, Python, Node.js

• DevOps: Continuous Integration & Continuous Deployment (CI/CD)

Myntra has multiple technologies spread across its platform. The ones listed above are the key ones, driving the e-commerce platform in every individual component. Each of these components is working flawlessly across the platform to offer users a smooth shopping experience.

Figure 1- Represents the diverse tech stack

**3. Comparison of SDLC Models**

**3.1 Waterfall Model**

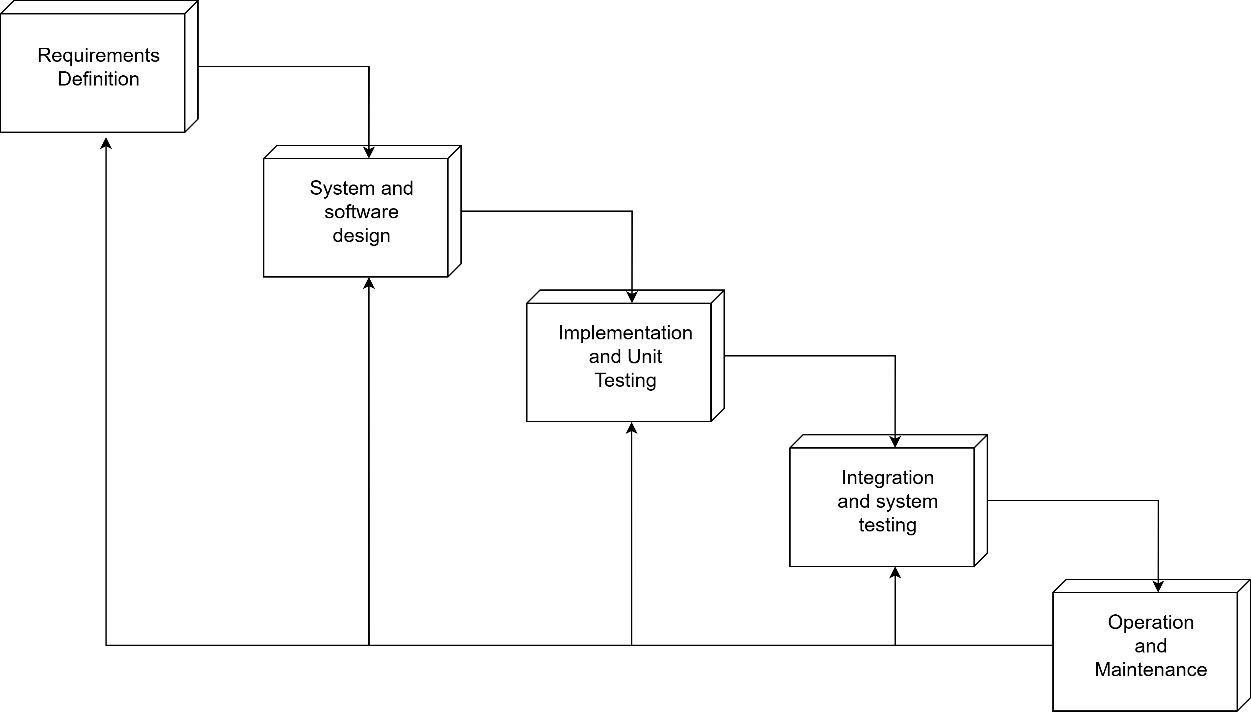
****

Figure 2- Illustrates the Waterfall Model

How Myntra Would Be Developed in Waterfall:

• Phase 1: Requirements Definition -- In this phase, all the requirements of the system are defined simultaneously. These include defining user login, product catalog management, payment gateway integration, and recommendation for personalization. Because it becomes difficult to make changes later, thorough documentation must be done at this stage.

• Phase 2: System & Software Design -- A full architecture is created during this phase, such as database schemas, API endpoints, and server infrastructure on AWS. Each component of the system is well-planned out prior to development.

•Phase 3: Implementation & Unit Testing -- The developers start coding the whole system at once, adhering to the architecture previously established. There are no modifications allowed to the requirements, and development is in a linear sequence.

•Phase 4: Integration and System Testing -- Once development is finished, the whole system is integrated and thoroughly tested. Functional testing, performance testing, and security testing are all performed. Bug fixing can take a long time.

• Phase 5: Operation and Maintenance -- The completely developed e-commerce platform is put into production. Any bugs or problems found after the launch are resolved during maintenance. Yet, updates are slow to deploy, which renders this model impractical for a dynamic company like Myntra.

Applicability to Myntra:

• Pros:

1. Well-defined and documented process.
2. Individual and clearly defined phases make it easier to manage the project.
3. Ideal for small, well-understood projects with low anticipated changes.

• Cons:

1. Lacks flexibility for rapidly changing user needs.
2. If issues are discovered later on, it can cause significant delays.
3. Long development cycles make it unsuitable for continuous updates.

Verdict: Not suitable, as Myntra requires frequent updates, rapid feature deployment, and adaptability to evolving user preferences.

* 1. **Incremental Development Model**

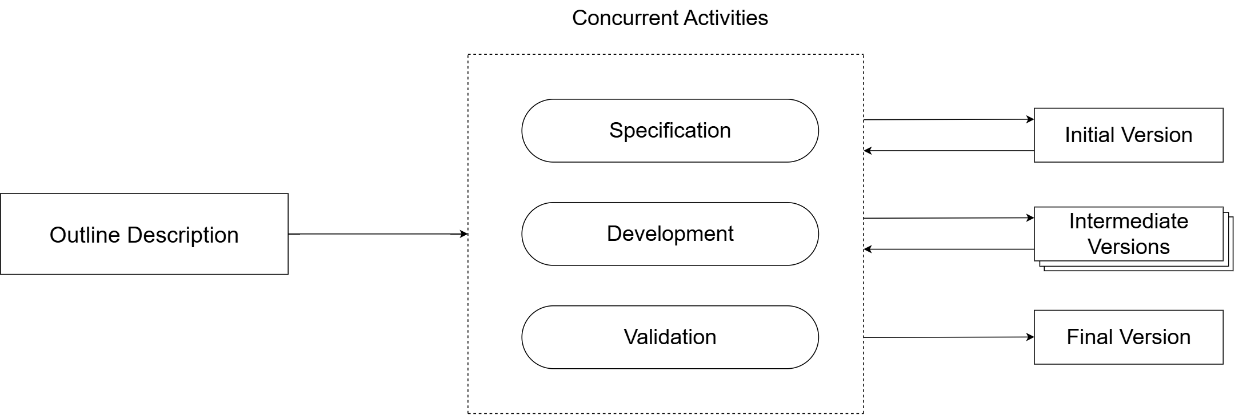
****

Figure 3- represents the Incremental Development Model

How Myntra Uses Incremental Development:

• Phase 1: Planning and Initial Requirements -- The development team determines high-priority features, including user authentication, product search, and payment gateway integration. Only the must-have requirements for the first iteration are determined, with scope for future updates.

• Phase 2: Feature-by-feature Development -- Rather than building the whole system at one time, Myntra regularly deploys new features. For instance, it can enhance the recommendation algorithms, introduce new product categories, and enhance the virtual try-on feature. Every feature goes through design, development, testing, and deployment in individual cycles.

•Phase 3: Ongoing User Feedback Integration -- Each increment is rolled out to beta users who give feedback through usage patterns and direct ratings. Developers review this feedback and make the required changes before releasing the next increment.

•Phase 4: System Testing and Refining -- All new features or updates are thoroughly tested, including performance analysis, bug fixing, and security checks. Automated pipelines guarantee ongoing integration and testing.

•Phase 5: Deployment and Monitoring -- Features are deployed to production in batches with minimal downtime. A/B testing is frequently employed to compare variations of a feature prior to full deployment.

•Phase 6: Iteration and Maintenance -- Engineers polish current features, fix user-reported bugs, and add new features based on changing user demands.

Suitability for Myntra:

**Advantages:**

* Increased time-to-market through incremental release of features.
* Improved responsiveness to changing user demands.
* Continuous testing ensures high reliability and better performance.
* Easily scalable and supports cloud-based microservices architecture.

**Cons:**

* Requires effective integration strategies to avoid system conflicts.
* Needs strong version control mechanisms to manage multiple development branches.
* Can become complex if too many features are developed simultaneously

Verdict: Highly suitable for a large-scale, evolving platform like Myntra, as it supports continuous and seamless updates and user-driven enhancements.

**3.3 Spiral Model**

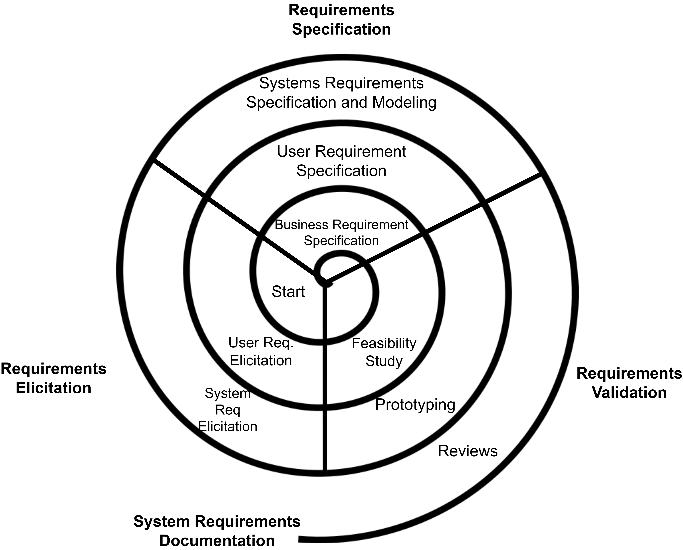
****

Figure 4- Represents the Spiral Model

How Myntra Would Be Built Using Spiral:

• Phase 1: Risk Analysis and Prototyping -- Before rolling out a major feature such as AI-driven recommendations, Myntra evaluates the risks involved in it. This covers biases in algorithms, threats to data security, and performance overhead. Small-scale prototypes are created before developing at a large scale.

• Phase 2: Concept Validation and Refinement -- After the prototype is tested, Myntra refines the feature design based on initial data. This iterative approach serves to validate the feature against technical constraints and business goals.

• Phase 3: Iterative Development and Testing -- The features are developed and tested in cycles. Developers implement changes based on risk assessment and early feedback from beta users prior to full-scale deployment.

• Phase 4: Wide Validation and Security Testing -- Features are created and built in numerous iterations on the basis of risk-based analysis and early feedback. Refinement is done, and the feature is rolled out incrementally to larger-scale environments until final deployment.

• Phase 5: Gradual Deployment and Performance Monitoring -- The developed feature is initially deployed to a few users. Real-world performance is monitored, and adjustments are made before scaling the deployment to a larger audience.

• Phase 6: Refinement and Continuous Improvement -- Developers use real-time analytics and user insights to refine the feature further. This cycle repeats for future improvements, ensuring innovation while minimizing risks.

**Suitability for Myntra:**

•Pros:

* 1. Robust risk management supports effective system building.
  2. Improvements iteratively refine advanced features such as AI-based personalization.
  3. Ideal for large-scale, high-risk functionalities like security patches and algorithm-based suggestions.

•Cons:

1. Costly and time-consuming for low-complexity features.
2. Highly skilled teams are needed to estimate risks effectively.
3. Ineffective for repetitive updates with no high-risk considerations.

• Verdict: Appropriate for high-risk features like security patches, AI-driven suggestions, and cloud infrastructure update, but possibly not required for regular feature development.

**3.4 Summary of Comparison**

| **SDLC Model** | **Flexibility** | **Risk Management** | **Time-to-Market** | **Cost** | **Suitability for Meesho** |
| --- | --- | --- | --- | --- | --- |
| Waterfall | Low | Low | Slow | Medium | Not Suitable |
| Incremental | High | Medium | Fast | High | Highly Suitable |
| Spiral | High | High | Moderate | Fast | Suitable for High-Risk Features |

**SDLC Flexibility Risk Time-to-Market Cost Suitability for**

**Model Management Myntra**

Waterfall Low Low Slow Medium Not Suitable

Incremental High Medium Fast High Highly Suitable

Spiral High High Moderate High Suitable for

High-Risk Features

**4**. **Requirements Engineering for Myntra**

Requirements engineering is essential for Myntra to achieve high performance, scalability, security, and a great user experience. This section discusses the key areas of its requirements engineering practices, their importance, and the challenges in managing them for such a big e-commerce site.

**4.1 Functional Requirements**

Functional requirements specify the particular behaviours and functions that Myntra needs to support in order to fulfill user expectations. These are:

•User Authentication & Account Management: Safe user sign-up and sign-in with OAuth. Multi-factor authentication (MFA) and social sign-in support. Several user accounts within one account with customized settings.

•Product Search & Navigation: Optimized search filters by categories, brands, and user interest. Customized UI/UX for an easy shopping experience.

•Payment & Order Management: Integration of various modes of payment such as credit/debit cards, UPI, and wallets. Automated order management and return processing.

•Recommendation System: Product recommendations using AI based on user behavior, history, and preference. A/B testing and machine learning models to dynamically adjust algorithms.

•Virtual Try-On & Augmented Reality: Virtual try-ons for accessories and clothing, with improved user experience.

•Content Management System: Backend user interface for managing catalogs, inventory, and promotions. System for new product listing approval and review.

•Customer Support: Chatbot and customer support system integration for live support.

•User Analytics & Insights: Aggregation of user engagement data for personalization and marketing purposes. Integration of predictive analytics to improve the shopping experience.

**4.2 Non-Functional Requirements**

Non-functional requirements establish the system's quality of operation and limitations.

• Scalability: The system should support millions of simultaneous users without any degradation in performance. AWS Auto Scaling and Load balancing can be utilized to dynamically scale resources according to demand.

• Availability: 99.99% uptime can be ensured through multi-region AWS deployment. Redundant servers to avoid outages.

• Security: End-to-end encryption, secure user authentication mechanisms, and fraud detection with regular security and penetration testing.

•Performance: Low product search latency and fast page loading. Utilization of Content Delivery Networks (CDNs) to provide faster content delivery.

•Compliance & Legal Considerations: Compliance with GDPR, data protection regulations, and consumer rights laws.

•Maintainability & Upgradability: Facilitation of continuous deployment and infrastructure updates.

**4.3 Requirements Validation Strategy**

Myntra utilizes numerous methods to validate software requirements:

•Stakeholder Reviews: Working with business teams, developers, and end-users to iterate requirements.

•Prototyping & A/B Testing: UI/UX testing at the earliest and decision-making on data about user behavior

•Automated Testing & CI/CD Pipelines: Implementation of unit, functional, and regression testing for conformity to a host of regulations.

•Security Audits: Continuous checking of vulnerabilities and penetration testing for conformity to security regulations.

•User Feedback Loops: Retrieving real-time analytics to comply with changing customer demands.

**4.4 Requirements Validation Challenges**

Even with an organized validation process, Myntra encounters difficulties in requirements validation:

•Dynamic User Expectations: Fashion trends and e-commerce have a changing dynamic, necessitating real-time adaptability.

•Global Compliance Issues: Complying with various regional content regulations and license agreements.

•Scalability & Performance Bottlenecks: Managing high traffic loads during sales and providing smooth service across varying network conditions.

•AI Bias in Personalization: Maintaining fairness and inclusivity in product recommendations.

• Fraud & Security Threats: Increasing threat of fraud and new risks in electronic payments and data security.

**5. Conclusion**

Myntra, being a constantly evolving online shopping portal, gains most from Incremental Development for swift deployment of features and Spiral Model for high-risk features. Waterfall model does not fit in as it has a rigid nature.

Myntra has a good requirements engineering process that ensures it satisfies both functional and non-functional requirements, giving its users a smooth shopping experience.

**7. References**

1.Sommerville, I. (2015). Software Engineering (10th Edition).

2.Myntra Technology Blog -

<https://tech.myntra.com>

3.Wikipedia –Myntra- https://en.wikipedia.org /wiki/Myntra