**Software Development Lifecycle (SDLC) – Analysis of Flipkart**

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

Avanthika U Achar

NNM23IS027

|  |
| --- |
| **Other keywords:** SDLC, Flipkart, AWS, Integration, Testing, Scalability, Requirements validation.  The software development domain continues to grow and evolved and therefore requires structured ways of building scalable, secure, and efficient systems. The report carried out will explore SDLC models applicable to Flipkart, a globally popular eCommerce system, and renders a comparative analysis of the SDLC methodologies and gives an overview concerning requirements engineering.  The study is said to provide a guide for choosing a suitable SDLC model for a management system and focuses on the waterfall, incremented development, and spiral model approaches. In addition, it goes on to discuss the challenges and strategies involved in requirements validation and software deployment at Flipkart.  The findings in this document are widely based on research, the industry best practices, and Flipkart website inputs. It is hoped that this report will serve as a valuable reference for software engineers, architects, and researchers who have an interest in the convergence of SDLC and requirements-engineering methodologies with large-scale streaming apparatus. Thus, the paper ends with a conclusion and a few considerations.  **Publishing:**  This paper was submitted to Dr. Jason Elroy Martis, Associate Professor, Department of Information Science and Engineering, NMAM Institute of Technology, Nitte Karnataka, India. This paper is also hosted on a GitHub repository, along with the material used for preparing this research. |

Table of Contents

1. Introduction 3
2. Analysis of SDLC Models 5-13

* 2.1 Waterfall Mode 5-7
* 2.2 Incremental Model 8-11
* 2.3 Spiral Model 11-13

1. Requirements Engineering Process 14-18

* 3.1 Functional Requirements 14-15
* 3.2 Non-Functional Requirements 15-16
* 3.3 Requirements Validation Strategy 17-18

1. Conclusion 19
2. References 19

**Introduction**

The paper provided a discussion of the concept of Product Life Cycle Management. And other concepts regarding the six stages happening in a Product Life Cycle: Introduction, Growth, Maturity, Saturation, and Decline.

**Development:**

In 2009, Flipkart began a basic website, selling $40 million in books across India. To scale its business, Flipkart managed to acquire $10 million and bought Weread, a book discovery platform. In 2012, Flipkart acquired electronics retailer Letsbuy, increasing its range of products. By 2013, sales hit record highs, making it Flipkart's best year in business.  
  
**Growth:**  
Flipkart has announced that, as far as the e-commerce in India is concerned, it had healthy growth till date capturing 31.8 market share in India, which remains said to be one of the quite high in the Indian market. Even started a platform for booksellers selling many of the books. These thoughtful ideas of them in turn would increase their growths at faster rates. Some of the very ideas which ultimately raises their growths are;

1.Acquisition and Expansion: Building any product from scratch takes lots of time and energy and blending into the concept. Post this stage, developing and testing formulate a big part of the growth phase of the product. Thus, flipkart started to acquire “weread”.  
  
2. Cash on Delivery: COD is one of the latest features that have made the payment convenience very much accessible for its customers. Because of this, the cart conversion rate has increased by 20 percent.  
  
3. Delivery: Flipkart acquired stakes in MapmyIndia, with a goal of improving options for deliveries. The last-mile delivery actually helped the customers receive an order within a few hours after the order had been placed. This delivery option helped in rural expansion.

4. Marketing: In 2011, Flipkart was acquired by Mime360, a content marketing company focused on the innovative strategies that propelled product sales.

1. **Analysis of SDLC Models**

A pictorial depiction of other SDLC models helps to articulate the various steps of software development, from planning to deployment. Models like the Waterfall, Incremental, and Spiral models support the developers' understanding of the software development process.

**2.1 Waterfall Model:**

In the Waterfall Model, the model is the paradigm of software development in all IT fields and generally is regarded the most basic of software development models. It is used when project goals are clearly defined, and such models usually apply to large-scale projects.  
Some of the phases of the waterfall model include requirement analysis, design, development, testing, deployment, and maintenance.

**2.1.1 Requirement Gathering**

The Flipkart business team works with stakeholders to identify these needs, goals, and platform functionalities. There may be a suggestion for new features like personalization based on customer feedback or suggestions for how to get a visitor through a finished payment process.

**2.1.2 System Design**

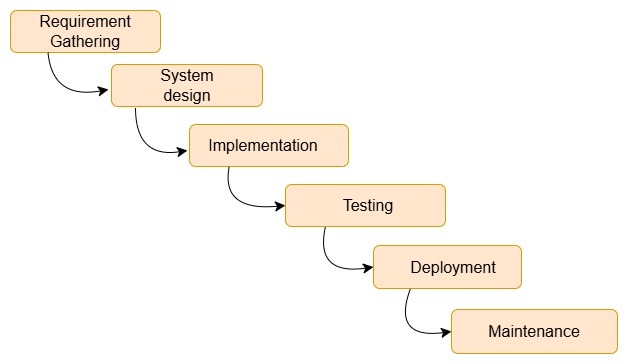
In this phase, Flipkart designers assume the responsibility of starting their work towards designing a proposed detailed solution according to the requirements; an architect to improve its recommendation engine, an architect to redesign a database schema to properly underpin the ever-increasing number of products and users.

**2.1.3 Implementation**  
The developers will be involved in writing codes and implementing features according to the design-say improved search filters or checkout flow. It is done on both UI/UX and backend sides.

**2.1.4 Testing**  
At Flipkart, the quality assurance team runs a barrage of tests-performance, usability, and security tests. They offer troubleshooting to diagnose and fix bugs and errors in new features, say the integration of payment gateways in the mobile app.

**2.1.5 Deployment**  
After rigorous testing, these features now come in readiness for deployment into production. This may also see them ushering into operation all the changes necessitated by Flipkart's website and native mobile applications.

**2.1.6 Maintenance**Upon rolling out those features, Flipkart engineers continue their monitoring of the platform so that they can quickly respond to bugs with immediate fixes, feed performance updates into it, and release routine upgrades to modernize the functionality and keep pace with evolving customer feature requests.

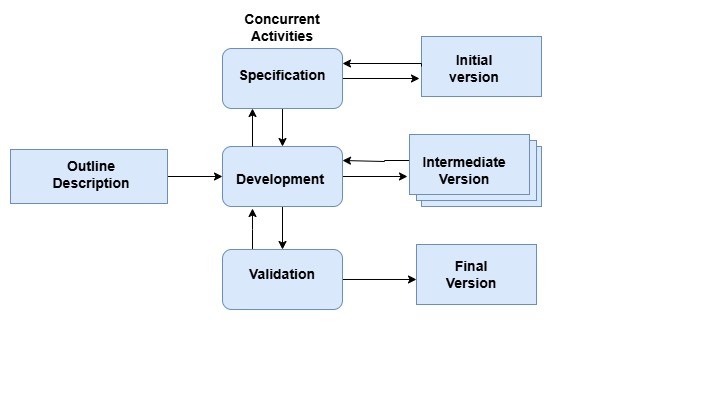


**Functional and Non-functional requirements:**

**Functional Requirements:** The waterfall model would be most suitable when distinctive functional requirements must be addressed by the developer. For example, in the case of Flipkart, this can be applied in cases such as the product catalogs and the payment mode, where requirements will be properly set from the beginning.  
  
**Non-functional requirements:** This model works better for non-functional requirements, which encompasses general quality attributes like performance, security, etc. Flipkart might specify such components as scalability, availability, or transactional capabilities.  
  
**Risk and Change Management:  
Risks:** The application of the waterfall model is delimited. Each of its utopias will only start when another will finish, and all will start after development is completed.

**Change Management:** The Change Management model makes for its high rigidity-there's an inability to go back to earlier phases after one phase of a project is completed. This could prove challenging for Flipkart, as e-commerce marketing and technology are evolving very fast. Updates can get confusing, and a good set of guidelines needs to be established to facilitate smooth change management.

**Time and Cost Constraints:**  
**Time:** This model is more slow in assisting due to the process furthering through set structural and sequential order.  
**Cost:** Surely, the attribute can be predictable. If we take an example, when the bug is discovered after it is deployed to the field can indicate change in costs. **2.2 Incremental Model:**An Incremental model is a model that divides itself into smaller segments or called increments. Each of smaller segments builds itself upon the last phase and is developed and tested thoroughly before it is moved on to the next one.

**Description of Outline:**  
In this first increment, we mostly scope the content of the entire system goals and define some of the needed features.  
Example from Flipkart: The initial increment focuses on the product catalog, user authentication, and simple checkout.  
  
**Specification:**  
After entering all the requirements, a more detailed specification for each of the functional specifications should be full of the safe scope for this purpose.  
If the first increment includes a product display system and user profile details.  
  
**Development:**During this phase, all agreed-on features from the specification are designed.  
An example of this is, such as user login and backend product catalog management, done at Flipkart.  
  
**Validation:**This shall comprise whatever further checking is done to find out whether some error existed in the produced software and will work correctly.  
For example, at Flipkart, for Usability purposes-the product catalog, and security and performance testing for the user authentication system.  
  
**Initial Version:**The first working software: only that increment's working features.  
Initial versions include: browsing a catalog and registering a user.  
  
**Intermediate version:**  
The following increments are simply adding one more iteration to the last: they either bring in new features to the Finish project or clean up a little of the mess the other versions left behind.  
Subsequent iterations added features to Flipkart's search page and shopping cart.  
  
**Final Version:**  
This means after everything's done and all integrations results in a full-fledged system having been tested and ready for deployment.  
In the example of Flipkart: A complete e-Commerce platform which is ready to be deployed developed search product checkout order tracking user reviews after several increments.**Functional and Non-functional requirements:**

**Functional requirements:** The incremental model comes into use when the complete functional requirement must be divided into smaller modules. This model is applied by Flipkart, for instance, for the In-Memory A/B testing for its users, allowing for features like product searching and reviewing features to be introduced first and then completely implemented and checked.

**Non-functional requirements:** Incremental model is applied for moderately addressing the non-functional requirements. Flipkart may start with minor features and then continue to improve them into an ensuing phase focused on continual improvement to enhance performance, data protection, and security.

**Risk and Change Management:**  
**Risk:** The incremental model reduces risk scenarios much more. Hence, Flipkart can ascertain any inefficacy of any of the earlier stages and recognize accordingly, hopefully making the necessary changes.

**Change Management:** During the lifetime of his project, any changes or modifications in it are handled as part of change management. Such changes include systematic control, review of changes that are about to be made to take care not to hinder the overall progress. Change management thereby becomes very important in incremental software development, where the system is built-up and upgraded through many little stages.

**Time and Cost Constraints:  
Time:** Rapid time-to-market is a key advantage this model brings to Flipkart. Flipkart can put out limited alternatives of their products into the marketplace in prior phases. Additionally, they can collect user feedback and refine this feedback into later phases, quickly bringing the market's value higher.

**Cost:** Cost for additional features and fixes can be accounted for to future iterations. Flipkart can focus on a few key resources at once and can adequately change its budget to account for performance fixes.

**2.3 Spiral Model:**A Spiral Model works iteratively on development and risk management by taking many spirals representing different products that ensure the risk probability and then this is analyzed at every iteration.

* + 1. **Planning:**

A primary goal of this phase is formalized and documented objectives for a project while constraints and high-level requirements can be specified.  
The detailed specification at Flipkart includes making improvements in the existing catalog and checkout system features with optimal scalability at peak loads during sales.

* + 1. **Analysis:**

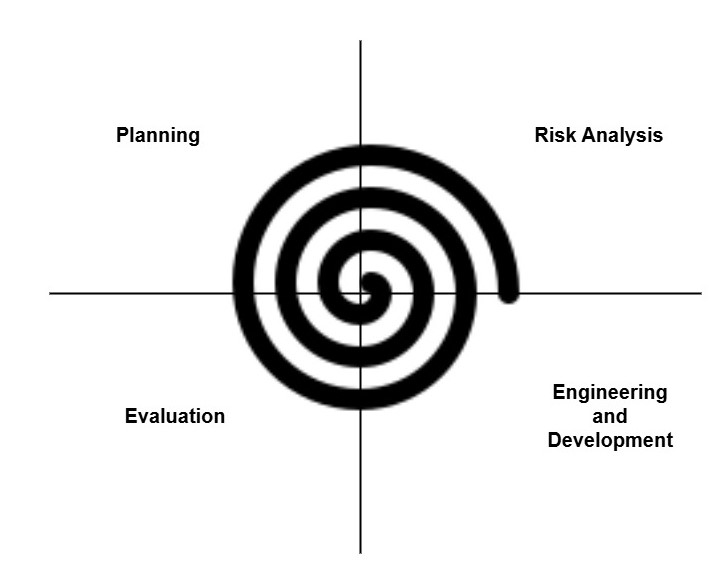
This analysis includes identifying and developing risk assessment and management plans.  
Some of the risks Flipkart plans to mitigate during the sales process include server overloads and failures of its payment gateways, by augmenting its infrastructure and conducting regular audits on security.

* + 1. **Engineering and Development:**

A kind of goal sought after from this stage remains the design and engineering of this specific increment to be built and tested.  
At Flipkart, the developers practically enhance product searches and the checkout system for security, support, and rapid performance of the system while allowing varied devices.

* + 1. **Evaluation:**

The ultimate goals of this stage include stakeholders reviewing all work and making adjustments for the next iteration.  
Flipkart reviews the feedback, performance, and reviews after releases of recent features into production, and modifies the next iteration to enhance product recommendations and the mobile interface.



**Functional and non-functional requirements:  
Functional Requirements:** This model mainly deals with highly complex projects, such as AI-Driven product recommendations, advanced systems and other intricate solutions in Flipkart, allowing the development of functional requirements within the product.  
  
**Non-Functional Requirements:** The vast scale of business transactions in Flipkart requires the development of non-functional specifications that include, among other things, scalability, performance, and security.

**Risk and Change Management:  
Risk:** The particular strong points in favor of spiral modeling are coming up with initiatives in terms of risk management. Losing focus on project risks, which include technical feasibility and resources allocation, could minimize the probability of major surprises each time Flipkart has to reassess them before the end of its iterations.

**Change Management**: Spiral modeling allows the managing of changes with utmost flexibility. With every revision of the working model, Flipkart has been able to assimilate user feedback quickly and hence easily modify features in line with changing market conditions or user preferences.

**Time and Cost constraints:**

**Time:** The total time of development is variable and greatly depends on the number of micro spirals undergone. It's good for Flipkart since multiple cycles will lead to quicker releases of smaller bug fixes or new features.

**Cost:** Costs in a Spiral model are difficult to predict due to a more iterative process; hence, the upfront risk assessments would help mitigate incurring huge costs in the later stages. However, for Flipkart, it can help escape the traps of wasting money on features that do not appeal much.

1. **Requirements Engineering Process**

This project describes the general requirements for creating an e-commerce platform for Flipkart, stating all the functional and non-functional requirements.  
  
This system is meant to provide an online marketplace for users to search for products, place products to the cart, and make purchases. The platform says that users may list the products and manage their inventory.  
  
It is designed to work for:  
- Registration, user login, and profile  
- Product Catalog with search and filter  
- Shopping cart management  
- Order processing, payments, and shipping  
- Sellers must have dashboards for inventory

* 1. **Functional Requirements:**

Essential functional requirements for a Flipkart e-commerce site are specified in this document, concentrating on user, product, order management, and seller functionalities.

1. User Registration & Authentication:

* Users can create an account with an email, phone number, and password.
* Users can log in using their credentials.
* Passwords can be retrieved via e-mail or SMS. Users can change profile information (address and payment details).

1. Product Catalog:

* Sellers can sell products with name, description, and price.
* Users can search and filter products.
* Users have a detailed product page which includes specifications, images, and reviews.

1. Shopping Cart:

* Users can add/remove products from the shopping cart.  
   The users should have access to view cart items and total.
* Checkout with delivery and payment details.

4. Order Management:

* Users would place, track, and cancel an order.
* Confirmation of orders with tracking info.

5. Reviews & Ratings:

* Users can review products (1-5 stars).
* Users can review the product.

6. Seller Dashboard:

* Seller manages all product listings (Add, update, and deletes).
* Sellers get notified on new orders.
* The seller checks sales, reports, and analytics.
  1. **Non-Functional Requirements:**

This document describes the non-functional parameters of the Flipkart e-commerce performance, support, security, scalability and other reliability and user experience factors

1. Performance:

* The system must have 1 million active users and there shouldn’t be any degradation in performance.
* Pages should load within two seconds during standard web traffic.
* The database should ideally respond in under 200 milliseconds.

1. Scalability:

* Vertical system scaling should be feasible in order to accommodate higher load during sales or festive seasons.
* Additional servers and resources should be added without negatively impacting user experience.

1. Availability:

* The system must adhere to the uptime guarantee of 99.9 percent for each year.
* Should any system failure arise, the system should have the ability perform disaster recovery within 4 hours in order to restore service.

1. Security:

* Users information such as passwords and card details must remain confidential and should be encrypted.
* Users depositing extra security by opting for two-factor authentication during log-in.
* Systems should be defended with security vulnerabilities like SQL injection, Cross Site Scripting (XSS) or even Cross-Site Request Forgery (CSRF).

1. Usability:

* Users should have responsive mobile, tablet and desktop interface to the platform.
* The system interface (UI) should achieve simplicity for users enabling easy navigation.
  1. **Requirements Validation Strategy:**

These guidelines comprise the following: setting explicit requirements reinforced for completeness, clarity, consistency, and feasibility.  
Stakeholder Engagement: Users should get involved as early as possible, through meetings and feedback sessions, together with business analysts and developers.  
Traceability Matrix: Ensure that each requirement returns to the business goals and user needs.

Walkthroughs and Reviews: Always make sure that requirements are being reviewed collectively with the team and stakeholders, where any issues can be addressed quickly.

Prototyping: Create models to elude some ambiguities and seek user participation.

Test Cases: Take utilization tests to determine whether requirements are achievable.

Testing Verification: Used during benchmarking of similar systems or against each industry's standard.

Formal Acceptance: Stakeholders concerned with the requirement sign the requirement document.

Some challenges may emerge:  
Ambiguous Requirements: Ambiguous requirements create confusion on understanding. Solution: these will be clarified with the stakeholders.

Conflicting Stakeholders: Different stakeholders demand different things, based on different interests. Solution: priorities should be determined and mediation steps should be taken.  
Changing Requirements: Requirements are fluid and tend toward extreme change. Solution: Support must be provided to ensure the change meteorologically follows established change control procedures.

**Conclusion**

The software development lifecycle (SDLC) at Flipkart is driving the delivery of scalable, efficient, and user-friendly features for buyers and sellers alike. Using flexible process models such as Spiral and Incremental allows e-commerce to keep reaping the advantages from Flipkart owing to continuous enhancement and possible changes in requirements. These strategies favor rapid releases, frequent updates, and sound risk management. This means Flipkart is fulfilling customer needs as they arise while fixing problems as they occur. Although the Waterfall model is itself orderly, it offers little scope for endless change commonly linked to e-commerce marketplace business.

**References**

1. Software-Engineering-9th-Edition-by-Ian-Sommerville.pdf -book
2. https://jitenderranga.medium.com/flipkart-through-product-lifecycle-8b597a57763a