PROJECT REPORT
ON
Software Development Lifecycle (SDLC) Analysis of Oracle Health A comparative study of different models in relation to Flipkart System Development

Submitted To

NMAM Institute of Technology, Nitte

(Off-Campus Centre, Nitte Deemed to be University, Nitte – 574110, Karnataka, India)

On partial fulfilment of the requirements for the award of the

Degree of Bachelor of Technology

In

INFORMATION SCIENCE AND ENGINEERING

By

Anchal Rao N

NNM23IS013

Under the guidance of

Dr. Jason Elroy Martis, Associate Professor, Department of Information Science and Technology, NMAM Institute of Technology, Nitte, Karnataka, India

PROBLEM STATEMENT

Your task is to analyse the software development lifecycle (SDLC) of a real-world system by conducting a comparative study of process models and their impact on requirements management.

1. **Case Study Selection:**

- Choose an existing large-scale software system (e.g., an e-commerce platform, a healthcare management system, or an ERP system).

2. **Analysis and Comparison:**

- Write a detailed report comparing Incremental Development, Spiral Model, and Waterfall Model for the chosen system. Highlight their suitability based on:
    - Functional and non-functional requirements.
    - Risk and change management.
    - Time and cost constraints.

3. Requirements Engineering Process:

- Develop a simplified requirements document for the system, including functional and non-functional requirements.
- Outline a strategy for Requirements Validation and identify potential challenges in this phase.

Table of Contents

Introduction

Understanding the Software Development Life Cycle (SDLC)

The Software Development Life Cycle (SDLC), is what one will have to follow in the process of software program development/maintenance in the area of software engineering. It provides a structured system for programmers to work in different phases-from planning and design through coding, testing, release, and maintenance. If they want to meet deadlines, compete on costs, and guarantee certain quality expectations, software teams must take an SDLC approach to their projects.

But there is no one best SDLC model; working into one type of model is dependent upon the particular needs of each new project. The common models include the Waterfall model, Incremental development, and Spiral model, offering various advantages based on the complexity of the project, its relative risk, and the availability of resources.

Large-scale systems need it to be defined right from the onset in terms of whether it is expandable, efficient, and maintainable in the long run. The success of such systems often depends on how well the development process is able to tailor itself to meet business needs, technology trends, and changing customer expectations.

Why SDLC is Essential for Large-Scale Systems

Large platforms, primarily e-commerce, must manage millions of daily users, high volumes of transactions, and real-time interactions in the present digital marketplace. Developers should not only build a working system but also ensure it is reliable, secure, and capable of adapting to market demands. For example, India's one of the eminent e-commerce platforms, Flipkart, seeks a strong development process to manage its operations in different realms, such as: scalability-the ability to support high traffic-primarily during flash sales and festive seasons-security and data protection-keeping customer data secure, preventing fraud, and safe processing of payments-regular development brought about due to constant change in innovations-a regular addition of features to give a better shopping experience and gain competitive advantage.

Failure to choose a proper SDLC model can result in delays in the project, system inefficiencies, or security vulnerabilities, all of which further dent user experience and affect the performance of enterprises. On the other side, a correct model allows Flipkart, among other improvements, to:

- Faster delivery of new features through iteration.
- Minimized risk exposure through regular checks into changes.
- Greater cost efficiency under high performance and secured conditions.

Thus, choosing the right SDLC model is not only a technical decision, but a strategic business requirement for large-scale platforms like Flipkart.

Purpose of This Report

This report aims to analyze and compare three most commonly used SDLC models-Incremental Development, Spiral Model, and Waterfall Model-in the light of Flipkart's software development. Such a comparison would guide in ascertaining the model (or combination of models) best suited for Flipkart's scale of e-commerce application.

This comparison will focus on:

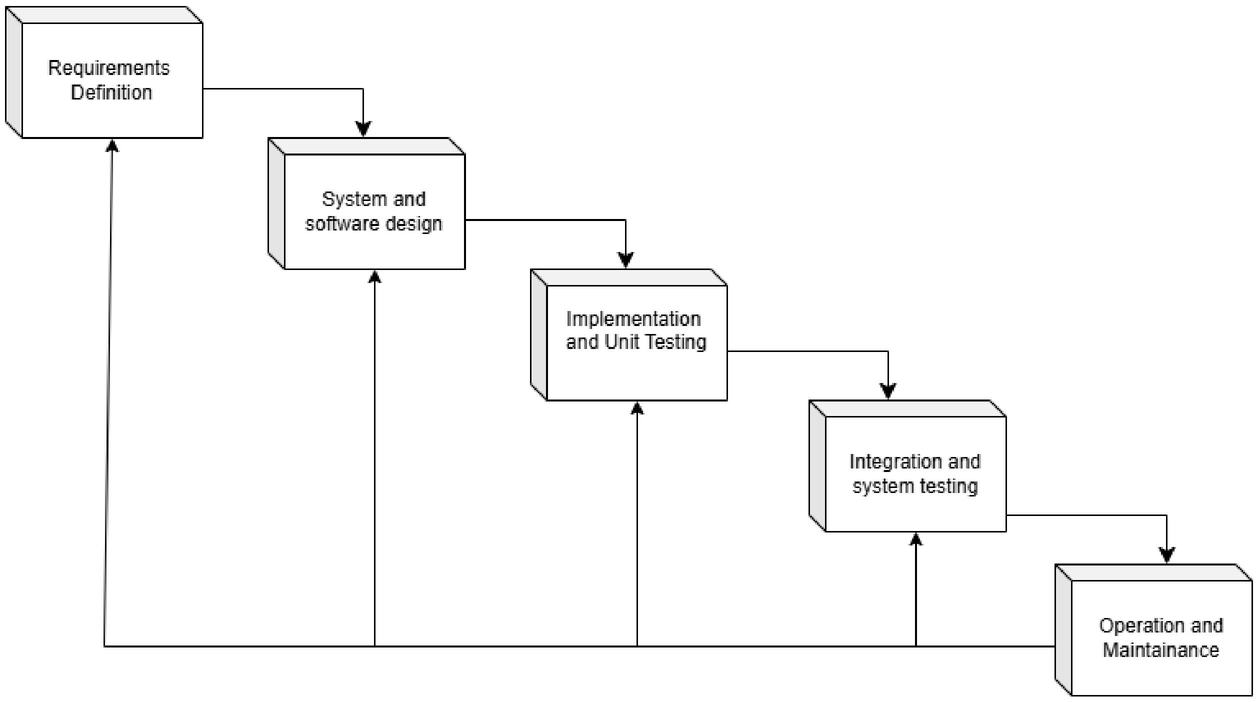
How each model supports functional and non-functional requirements.

- The effectiveness of each model in the risk management stage.
- The assessment of time and cost efficiency for large-scale software projects.
- A recommendation for the best suitable SDLC approach for Flipkart.

Through this, we seek to understand the influence of SDLC decisions on software quality, business agility, and long-term system sustainability in real-world e-commerce applications.

Comparative Analysis of SDLC Models

1.Waterfall Model


Requirements Definition
System and software design
Implementation and Unit Testing
Integration and system testing
Operation and Maintainance

Phase 1:*Requirement Definitions Regular:*

Identify core e-commerce functionalities-product listings; cart; checkout; payments, etc.

Define the scalabilities, security, and performance requirements.

Document the third-party integrations such as logistics, payments, and analytics.

Ensure all regulatory and compliance requirements are adhered to.

Thus the outcome expects a Software Requirement Specification/SRS document.

Phase 2:*System/Solution Design*

Design data architecture for products, users, and orders.

Define system architecture: microservices, cloud, load balancer.

Build UI/UX based wireframes for smooth shopping trips.

Select the tech stack for Java, Python, React, MySQL, AWS.

Put in security like SSL, encryption, and two-factor authentication.

Thus the outcome will yield a system architecture, database schema, UI wireframes, and technical documentation.

Phase 3:*Implementation and Unit Testing*

Build the front end (React, HTML, CSS, JavaScript) and for the back (Node.js, Python, and Java).

Build key modules:

Product Catalog (search, categories, filtering).

User Management (sign up, login, profiles).

Shopping Cart & Checkout-add/remove items, pricing.

Payment Animal-Integration with UPI, cards, Flipkart Pay Later.

Logistics & Order Tracking-integration with Ekart.

And conduct unit testing to verify separate components.

That is, the outcome expects the first working version of Flipkart with core base functionalities.

Phase 4:*Integration and system testing*

Integration testing will check interactions across all modules.

Load testing: simulate peak traffic such as BBD.

Security testing: checking for data protection vulnerabilities and gateway interfaces for payments.

UAT aims at users to give real feedback to improve detection.

Thereby the outcome is a fully tested, stable, secured e-commerce system that could deploy in real time.

Phase 5:*Deployment & maintenance*

Implementation on the cloud.Bit and Googles constitute another cloud deployment task.

System health monitoring with custom reporting tools and mechanism for addressing and responding to performance issues real-time.

Regular security patching and periodic updates in functionality.

Provisioning for customer support services for insights and issue resolution.

Suitability to our system

1) Pros

- *Clear and Organized Proces*- Each phase is well-defined and comprehensively documented, which allows Flipkart to be less dependent on people to cope with complexity. Of course, everything comes with its own price--in particular time--so it also does require more complex coordination and communication between the teams involved.
- *Thorough Planning* - Since all different parts(e.g., product listings, payments, and order tracking) and all major features are reached in agreement from the very beginning, there is less likelihood that confusion could exist
- *Smooth Coordination*- Different teams design their own standards of coordination, communication, resource allocation, and so forth depending on the phase.
- *Great for Large Scale Developmen*t –Because it starts addressing security,
- Scalability, and compliance from the beginning, this is essential for a platform as, large as Flipkart.

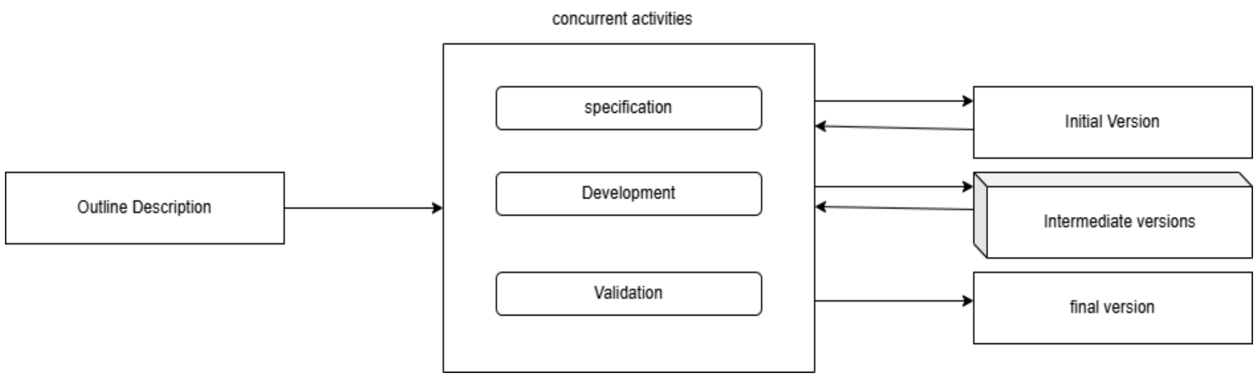
2) Cons

- *Not Flexible* -Because market trends or business needs could change, it is difficult to make appropriate changes to the system without being able to go back and rebuild previous phases.
- *Late Bug Detection* -Things like slow checkout or payment failures often become apparent only at the very end, these problems can be harder to fix..
- *Expensive and Time-Consuming Changes*- Whenever a revisal occurs after a development, it means a rollback into some previously done work, and that takes time and money.
- *Not Ideal for Ongoing Improvements*- Flipkart, by its nature, always adds changes by way of new features or updates. A case presumed, it does not provide an easy avenue for process control.
- *Better for the First Build, Not for Growth* -Although it works favorably for starting the platform piece, its invincible for continuous innovations that a more flexible methodology, such as Agile better suits.

Final verdict

Flipkart is not suited to the Waterfall model. An e-commerce setup is subjected to continuous evolution due to changes in market trends, customer preferences, and technological developments. The rigidity of this model makes it difficult to accommodate changes once the development starts. Moreover, Flipkart also relies heavily on continuous feedback from users, sellers, and logistics partners, which would not be feasible under the Waterfall model's late-stage testing approach. An Agile, iterative, and flexible model would ensure better handling of dynamic business needs.

<u>Incremental Development Model</u>



Phase 1:*Building the Basics of the Platform*

Flipkart came up with the most crucial functions that are needed for an online marketplace. These are product browsing, user registration, carting, and a basic checkout process. A secure payment gateway is integrated into a basic logistics system dealing with order processing and deliveries. This step ensures that the platform is completely functional and prepared for consumers to begin purchasing.

Phase 2:*Making Shopping a Joyful Experience*

Once the basic system is running, the next step is to enhance the experience for the user. They want features like wishlists, suggestions based on preferences, customer reviews, and other things to make shopping more interactive and customized. A mobile app is launched to make it easier for users to shop on the go, and the design of the website is enhanced for smoother navigation.

Phase 3:*Security and Payment Focus*

With the increase in users on the platform, security became a priority. Flipkart enhanced its payment range by adding UPI, digital wallets, and EMI facilities. Concurrently fraud detection systems were put in place to identify suspicious transactions. A completely seamless return and refund method came in to win over customers and ensure them they would feel secure with regards to purchasing through this medium.

Phase 4:*Expanding to Third-Party Sellers*

Where Flipkart has begun to sell products, it tumbles down, however, to become more of a mature marketplace that could host other sellers for listing and selling their products. A seller dashboard is developed to help merchants manage their inventory, track orders, and analyze sales. Having assessed all those ratings and reviews, users could have customer assurance regarding products and sellers on the platform.

Phase 5:*Enabling High Traffic and Scaling from There*

With an increasing number of users, checks launch the system on a proper scale to harness its power for their benefit. After its launch, help crashed on Flipkart's platform while being optimized to manage millions of users during tips-trips such as Big Billion Days. To avert server crashes and slow operating time, the team utilizes a blend of cloud-based solutions, load balancing, and AI supply-chain operation systems. They also incorporate live tracking for exact delivery date updates.

Phase 6:*Continuing Innovations and New Features*

Advanced features balance out wealth buildup and results for Flipkart: voice search, AI-assisted customer support, and one-day delivery ease customer access. It has begun taking sustainability initiatives in the form of green packaging and efficient logistics.

Suitability to our system

1) Pros:

- There are infinite possibilities in the flexible and adaptable nature of Flipkart because it can roll out new features in phases making it easier to adjust to the changing customer needs and market trends.
- *The core features can be launched quickly*: in such a platform, not everything must be in place for launch. Functions like browsing products and checking out can go live earlier.
- *Less risk:* as procedures are done step by step, any problems or bugs, as they arise, can be fixed immediately, thus preventing dovetailing into larger issues later on.
- *Develop cost-effectively:* Factored for maximum usage of resources, with attention on what's most important and no time wasted on developing features that may not be necessary.

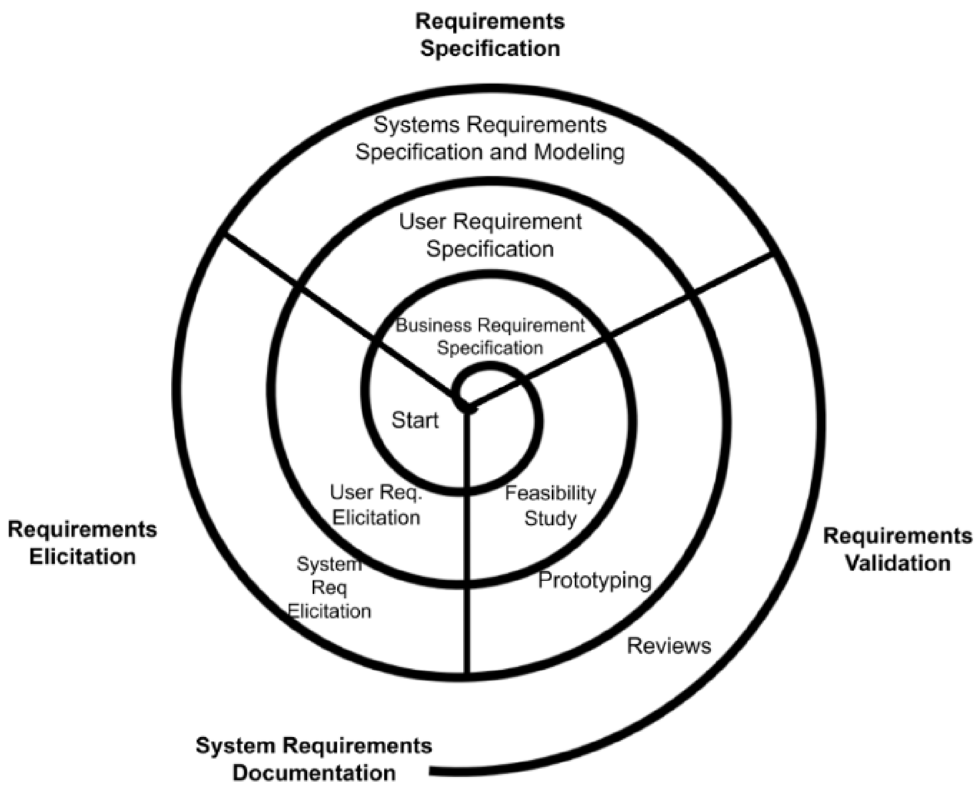
2) Cons:

- *One Phase Depends on Another:* Some features may not be able to be built until the completion of other features and this may delay the progress.
- *Integrations are Tough in Complicated Ways:* Sometimes it requires extra testing to add features to a system that is already running.
- *Ongoing Maintenance Effort: Every* update needs to be tested, repaired, and optimized, which adds to the developers' workload.
- *Potential:* While the platform can go live in advance, it's a life-long task to round off all the rich features and enhancements.

Final verdict

The Incremental Model presents itself as a better alternative to the Waterfall Model for Flipkart but comes trapped in its frills. While it facilitates incremental development and early delivery of core functionalities, it may find it difficult to efficiently incorporate continuous feedback from users and stakeholders. Implementing any significant system-wide changes may become challenging, particularly if a myriad of changes has to be coordinated across several increments. An Agile or a Spiral approach might be more adaptable to the constantly changing business environment at Flipkart.

<u>Spiral model:</u>


Requirements Specification
Systems Requirements Specification and Modeling
User Requirement Specification
Business Requirement Specification
Start
User Req. Elicitation
Feasibility Study
System Req Elicitation
Prototyping
Reviews
Requirements Elicitation
Requirements Validation
System Requirements Documentation

Step 1:*Planning and Requirement Understanding*

In the beginning of every cycle, Flipkart's team collects vital requirements and analyzes them for functionalities- such as product listing, payment options, and order tracking options. This phase will also be re-visited at each iteration to refine and revise requirements, especially considering how customer expectations and market trends shift over time.

Step 2*:Risk Management and Prototyping*

Identifying and assessing risks before getting into the real development phase is crucial. Possible risks include anything from security threats to payment failures to millions of transactions the application must handle on the day of Big Sales. To prevent most of these risks from happening, and to test the application mechanics in advance, some kind of prototype should be created.

The process could involve designing a primitive version of Flipkart's shopping cart or checkout system to be tested for bugs and problems before serious resources are deployed.

Step 3:*Development and Testing in Small Increments*

After the risks are identified and reduced, the real development work begins. Flipkart builds each chance for stability in small incremental modules rather than a fully developed platform. These modules may include user registration, the product lookup, or placement of orders. Each module is extensively tested to ensure smooth functionality, so when it comes time to tackle another module, it is far easier to catch bugs and fix them.

Step 4:*Launch and Customer Feedback*

After successful testing, the updates are released to a limited group of individuals similar to a beta version. This gives Flipkart a good chance of receiving real-time feedback from the customers and to ascertain how the new features perform in real-time. Resolutions to help customers can be made before a wider version is available.

Step 5:*Continual Improvement through Expansion*

The process never stops at the time of deployment. Verification of further improvements in features is still possible on the part of Flipkart through user feedback and growing market trends. Each cycle takes a previous cycle as a base for deploying the functionalities of the platform on a growing basis such as AI-enabled recommendations, fast checkout options, and enhanced fraud detection systems.

Suitability to our system

1) pros:

- *Management of Project Risks through its Integration* – With Flipkart being an e-commerce platform easily beset by issues like threats to cybersecurity and server scalability, the Spiral Model seeks to take into account and address risks that may set out at the very start.
- *Flexibility of Requirements* – The Spiral Model is less rigid in its approach to customer and business needs than other models, but allows Flipkart to change its face each time the customer needs something.
- Early Prototyping – Leaving prototypes functioning during every phase makes sure that services like payment gateways and order tracking are tested repetitively before large-scale development.
- Continuous Customer Feedback – Features deployed through phases allow Flipkart to draw on user feedback from the get-go and involve appropriate improvements between iterations.

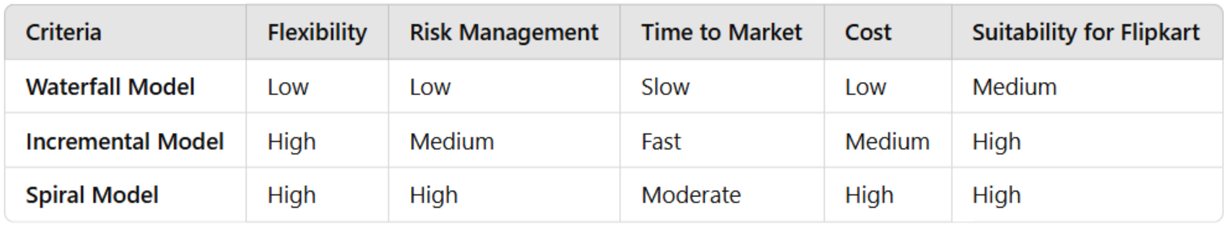
2) Cons

- *Expensive and Lengthy*-Continuous iteration, risk assessment, and testing with higher frequency make this one much more expensive and time-sensitive than traditional methods.
- *Complex Implementation*-Multiple iterations with risk assessments are usually under the direction and management of experienced teams under strong coordination, which is why the process tends to be quite complex.
- *Not Suitable for Minor*-Scale Features-If for a small update such as a UI improvement, then the spiral model seems too resourced for such little updates.
- *Requires Extensive Documentation*-There is a need for adequate documentation of risk, changes, and feedback to be kept, as the requirements keep changing quite frequently.
- *Potential of Open-Ended Development*-If not managed well, the continuous iteration process can lead to a situation of always expanding the goal of the project without an identified completion date.

Final verdict

Spiral is a better choice than Waterfall and Incremental models for Flipkart. The paradigm combines iterative development with risk assessment and therefore is best suited to deal with changing requirements in enterprise e-commerce. With re-prototyping going on continuously and stakeholder feedback taken at every stage, Flipkart will be in a better position to adapt itself to market changes, enhance security, and keep scalability under control. However, the Spiral model can be resource-heavy and challenging to manage. The Super Spiral model provides a blend of structured planning and adaptability when implemented successfully and becomes a viable option for overcome Flipkart's emerging business requirements.

Summary of all the models



Criteria	Flexibility	Risk Management	Time to Market	Cost	Suitability for Flipkart
Waterfall Model	Low	Low	Slow	Low	Medium
Incremental Model	High	Medium	Fast	Medium	High
Spiral Model	High	High	Moderate	High	High

Functional Requirements

Functional requirements are requirements describing the specific functions and operations that Flipkart must provide to ensure errors in e-commerce performance.

i. *User functions*
   - user registration, login, and authentication.
   - profile management (addresses, payment methods, preferences).
   - role-based access (customers, sellers, delivery partners, admin).

ii. *Product & Inventory Management*
   - product catalog with category, images, description, and specifications.
   - search and filter (by brand, price, ratings, etc.).
   - product recommendations based on user behavior (AI-based).
   - dynamic pricing and discount management.
   - inventory control and stock update.

iii. *Order & Payment Processing*
   - functions of shopping cart and wishlist.
   - multiple payment options (Credit/Debit, UPI, Wallets, COD).
   - order confirmation, invoicing, and alerting system.
   - order tracking.
   - return, refund, and exchange management.

iv. *Logistics & Delivery Management*
   - automated assignment of orders to delivery partners.
   - deliveries on schedule and ETA.
   - live tracking of orders and route optimization.
   - Cash on delivery.

Non-Functional Requirements

Non-functional requirements require particularly the quality of the characteristics and aspects concerned with the beholder of the system, performance, and security.

i. *Performance & Scalability*
   - Must support millions of concurrent users and transactions.
   - Search response time should be <2 seconds.
   - Well-scalable solution with a cloud-based infrastructure to handle peak loads (e.g., Big Billion).

ii. *Security & Data Protection*
   - Secure login authentication (2FA, encryption).
   - PCI-DSS compliant security around payment.
   - Protection against SQL Injection, XSS, and other cyber threats.
   - Data privacy compliance (GDPR, IT Act 2000).

iii. *Availability & Reliability*
   - 99.99% availability with failover mechanisms.
   - Automatic backup systems to prevent data loss.
   - Load balancing.

Requirement Validation Techniques

1. *Prototyping*

   • Process-flow prototypes/mockups for key workflows (product search, checkout, seller dashboard).

   • Collect user feedback for UI/UX enhancements.

2. *Stakeholder Reviews*

   • Hold review meetings involving customers, sellers, logistics teams, and business leaders.

   • Confirm whether business objectives align with requirements.

3. *Use Case & Scenario Testing*

   • Develop real-world use cases (order placing, canceling a product, handling peak sales).

   • Ensure that system behavior is as expected.

4. *Requirement Walkthrough and Inspection*

   • The formal review of the Software Requirement Specification (SRS).

   • Check for inconsistencies, ambiguities, and missing requirements.

Conclusion

Based on the analysis among all those used - Waterfall, Incremental, and Spiral - ***the Spiral model is the most suitable for Flipkart***, combining iterative development with risk management, continuous feedback, and flexibility to meet fluctuating market trends.

- Waterfall is rigorous for Flipkart's evolvement.
- Incremental paves the way for keen phased development but rather struggles during update implementation on a huge scale.
- Spiral provides the required flexibility in risk evaluation and better adaptability, proving to be the best-fit choice for Flipkart.

Citations

1. Geeks for geeks Software Development Models : https://www.geeksforgeeks.org/top-8-software-development-models-used-in-industry/

2.Draw.io :

http://draw.io

3.Flipkart

https://www.flipkart.com