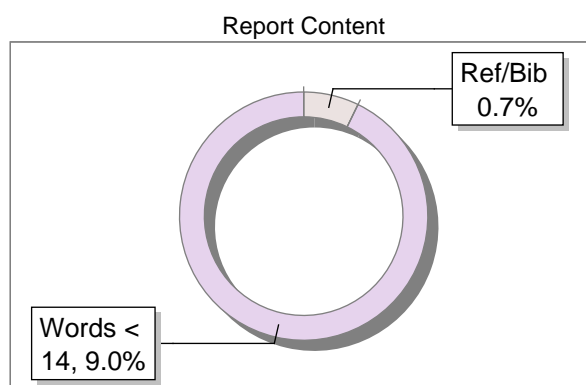
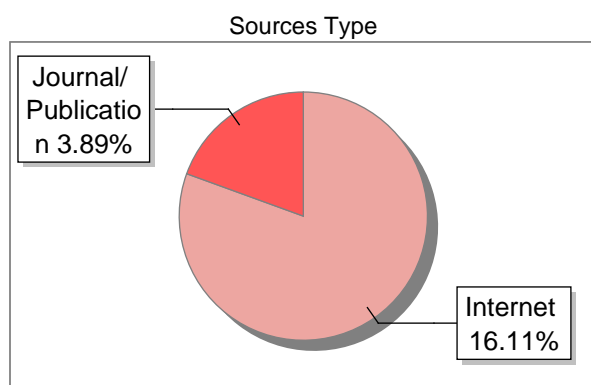


Submission Information

| | |
|--------------------------|----------------------|
| Author Name | Bharath D nayak |
| Title | Task |
| Paper/Submission ID | 3335282 |
| Submitted by | nnm23is030@nmamit.in |
| Submission Date | 2025-02-16 08:10:02 |
| Total Pages, Total Words | 16, 3577 |
| Document type | Project Work |

Result Information

Similarity **20 %**



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 Similarity Report

20

SIMILARITY %

39

MATCHED SOURCES

B

GRADE

A-Satisfactory (0-10%)

B-Upgrade (11-40%)

C-Poor (41-60%)

D-Unacceptable (61-100%)

| LOCATION | MATCHED DOMAIN | % | SOURCE TYPE |
|----------|--|----|---------------|
| 1 | www.intellectsoft.net | 4 | Internet Data |
| 2 | aws.amazon.com | 4 | Internet Data |
| 3 | careers.l3harris.com | 1 | Internet Data |
| 4 | frontiersin.org | 1 | Internet Data |
| 5 | intechhouse.com | 1 | Internet Data |
| 6 | lavallo.pl | <1 | Publication |
| 7 | www.askamanager.org | <1 | Internet Data |
| 8 | archiv.ub.uni-heidelberg.de | <1 | Publication |
| 9 | documents.mx | <1 | Internet Data |
| 10 | qdoc.tips | <1 | Internet Data |
| 11 | www.deskera.com | <1 | Internet Data |
| 12 | An operational effectiveness analysis of legacy and future mine count, by Beery, Paul Byram,- 2019 | <1 | Publication |
| 13 | eksss.org | <1 | Internet Data |
| 14 | sci-int.com | <1 | Publication |

| | | | |
|----|--|----|---------------|
| 15 | An empirical analysis to assess the determinants of SaaS diffusion in, by Martins, Ricardo O- 2016 | <1 | Publication |
| 16 | docplayer.net | <1 | Internet Data |
| 17 | ijcsn.org | <1 | Publication |
| 18 | moam.info | <1 | Internet Data |
| 19 | Optimal Path Search for Recurrence Relation in Cartesian Gaussian Integrals by Liu-2016 | <1 | Publication |
| 20 | search.emarefa.net | <1 | Internet Data |
| 21 | www.readbag.com | <1 | Internet Data |
| 22 | Assessment of cancer clusters using limited cohort data with spreadsheets Appli by Alla-1994 | <1 | Publication |
| 23 | devstars.com | <1 | Internet Data |
| 24 | energyenvsci.org | <1 | Internet Data |
| 25 | gcnagda.org | <1 | Publication |
| 26 | journal.unnes.ac.id | <1 | Publication |
| 27 | kaptatravel.com | <1 | Internet Data |
| 28 | mdpi.comjournalacoustics | <1 | Internet Data |
| 29 | pdf4pro.com | <1 | Internet Data |
| 30 | pdfcookie.com | <1 | Internet Data |
| 31 | smsdocs.aquaveo.com | <1 | Publication |
| 32 | Three-dimensional flow and load characteristics of flexible revolving wings by va-2018 | <1 | Publication |

| | | | |
|----|--|----|---------------|
| 33 | www.alliancetechpartners.com | <1 | Internet Data |
| 34 | www.aut.ac.nz | <1 | Publication |
| 35 | www.cubesoftware.com | <1 | Internet Data |
| 36 | www.geeksforgeeks.org | <1 | Internet Data |
| 37 | www.indianretailer.com | <1 | Internet Data |
| 38 | www.mageplaza.com | <1 | Internet Data |
| 39 | www.synergita.com | <1 | Internet Data |

PROJECT REPORT

ON

Software Development Lifecycle (SDLC) Analysis of Oracle Health A
comparative study of different models in relation to Oracle Healthcare 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

Bharath D Nayak

NNM23IS033

Under the guidance of

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

1.Introduction

Flipkart, one of India's biggest e-commerce giants, ²⁸ has completely transformed how people shop online. From electronics and fashion to groceries and furniture, Flipkart has become a one-stop shop for millions of users. Over the years, it's grown into a massive platform, and its success is a testament to its ability to innovate and adapt to customer needs.

This project dives into how Flipkart builds and maintains its platform, focusing on its requirements engineering process and implementation strategies. The main goal is to explore three different development methodologies: the Waterfall Model, the Spiral Model, and the Incremental Development Model. We'll look at key aspects like functional and non-functional requirements, risk management, change management, and time and cost constraints. The idea is to figure out ³⁹ which approach works best for a fast-paced, ever-evolving platform like Flipkart.

When you're dealing with millions of users, high traffic during sales events, and constant updates, the stakes are high. Flipkart needs a development model that balances speed, security, and scalability. This analysis will help us understand how each model addresses challenges like handling peak traffic, integrating new features, and ensuring data security—all while keeping the platform reliable and user-friendly.

The choice of development methodology is a big deal for Flipkart. It's ⁸ not just about rolling out new features quickly; it's also about making sure the platform is secure, scalable, and compliant with regulations. ²⁰ This research will shed light on how Flipkart can stay ahead in the competitive e-commerce space while maintaining customer trust and delivering a seamless shopping experience.

This report is super relevant, especially as e-commerce continues to grow and evolve. ¹³ The findings will offer valuable insights into the ²⁹ best practices for software development in the e-commerce sector, helping platforms like Flipkart stay agile, secure, and customer-focused in a rapidly changing digital landscape.

1.1 What is SDLC:

The Software Development Life Cycle (SDLC) is basically the step-by-step process of planning, building, testing, and rolling out a software system. Picking the right SDLC model is super important, especially for big projects like Flipkart, which is one of India's top e-commerce platforms. The whole point of using an SDLC model is to make sure we end up with high-quality, easy-to-maintain software that actually does what users need it to do. These models give us a clear plan for each stage of development, so everything runs smoothly, stays on budget, and gets done on time—all while making sure the final product hits the mark for users.

1.2 Why is SDLC important:

Software development can be challenging to manage due to changing requirements, technology upgrades, and cross-functional collaboration. The software development lifecycle (SDLC) methodology provides a systematic management framework with specific deliverables at every stage of the software development process. As a result, all stakeholders agree on software development goals and requirements upfront and also have a plan to achieve those goals.

Some benefits of SDLC:

- Increased visibility of the development process for all stakeholders involved
- Efficient estimation, planning, and scheduling
- Improved risk management and cost estimation
- Systematic software delivery and better customer satisfaction

1.3 How does SDLC work:

It does several tasks required to build a software application. The development process progresses through multiple stages, during which developers implement new features and resolve software bugs.



Stage 1: Planning & Requirements Analysis

At the beginning of the software development life cycle, one of the most critical steps is assembling a skilled and experienced team of engineers. These professionals play a key role in understanding the client's needs and laying a strong foundation for the rest of the project.

This initial phase involves several important steps. First, the team conducts a thorough analysis to identify the client's goals, challenges, and requirements. Based on this analysis, they propose a range of potential solutions, each accompanied by a detailed budget and plan. This ensures that the client has a clear understanding of their options before moving forward.

Stage 2: Defining Requirements

The next stage in the software development life cycle involves a detailed exploration of the requirements, once the client has selected a preferred software solution. During this phase, the team carefully reviews all project-related documents and assesses the client's existing systems and environment. This stage is often seen as a bridge between the initial planning and the more in-depth requirements analysis, as it combines elements of both to ensure a smooth transition into the development process.

Stage 3: Designing Product Architecture

After thoroughly analyzing the client's requirements in the earlier stages of the software development life cycle, the developers move on to designing multiple product architectures. These designs are presented to the client, who selects the one that best fits their needs.

Once the preferred architecture is chosen, it is formally documented in a Design Document Specification (DDS). This document is then reviewed by all stakeholders to evaluate key factors such as risk, reliability, flexibility, and cost-effectiveness. The finalized architecture serves as the blueprint for all subsequent stages of the software development life cycle and forms the core foundation of the final product.

Stage 4: Developing the Product

This stage is one of the most critical phases in the software development life cycle, as it focuses on turning ideas into working code and delivering tangible results to the client. Development typically takes up the majority of the project's timeline, and it's not uncommon for the process to extend beyond the initially estimated deadlines. This can happen for various reasons, such as the client requesting additional features or changes to the project scope midway through development. In some SDLC models, the product itself may evolve significantly during this phase, adapting to new requirements or insights as the work progresses.

Stage 5: Testing the Product

Once all the groundwork from the previous stages of the software development life cycle is complete, the focus shifts to testing. Quality assurance engineers step in to identify and track down any bugs or issues in the system. Testing is a critical step in the SDLC because it helps catch and resolve major problems before they can cause significant damage or losses.

While testing often occurs throughout various stages of development, it also stands as a dedicated phase. During this stage, the team creates a detailed map of all the issues and breakdowns that need to be addressed, ensuring the final product is as polished and functional as possible.

Stage 6: Deployment & Maintenance

Once the product is ready for release, the next step in the software development life cycle is to present the final software solution to the client. This marks the culmination of the development process, where the team showcases the completed product and ensures it meets the client's expectations.

To ensure the digital solution continues to function smoothly in the long term, the client can opt for ongoing product maintenance after all the SDLC phases are complete. This maintenance phase addresses any issues that may arise and keeps the software up-to-date, effectively bringing the software development life cycle to its conclusion.

1.4 What are SDLC models:

The SDLC model provides a structured framework to help organizations implement the software development life cycle effectively. Different models arrange the phases of the SDLC in various orders to optimize the development process based on specific project needs. Below, we explore some of the most popular SDLC models:

- **Waterfall Model:** A linear and sequential approach where each phase is completed before moving to the next.
- **Incremental Development Model:** The product is developed in small, manageable increments, with each increment adding new functionality.
- **Iterative Model:** The project is broken into smaller iterations, allowing for continuous refinement and improvement.
- **Spiral Model:** Combines iterative development with risk analysis, making it ideal for complex projects.
- **Agile Model:** Focuses on flexibility, collaboration, and delivering working software in short, iterative cycles.

For Flipkart, a combination of the Incremental Development Model and the Spiral Model is used. This hybrid approach allows for gradual feature development while incorporating risk assessment and iterative improvements, ensuring a robust and scalable solution.

2. Analysis of Incremental Development and spiral Model for Flipkart

First let us know about waterfall model, incremental development model and Spiral Model with their differences.

2.1 Waterfall Model

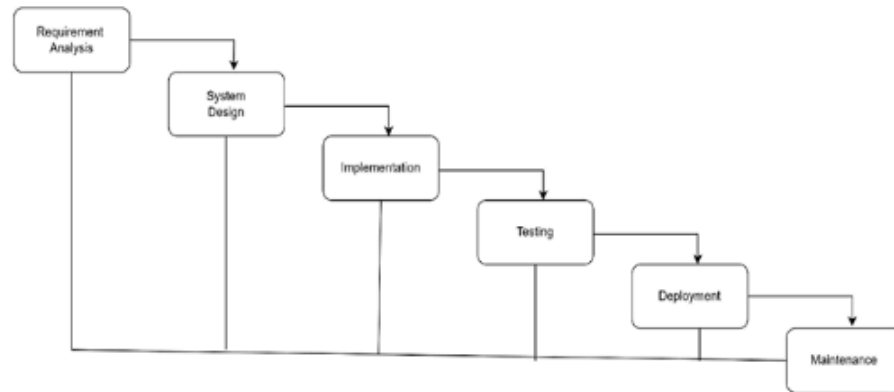
The Waterfall Model is also known as the Classical or Traditional Model. Sometimes, it is referred to as a linear-sequential lifecycle because, in this model, each phase is completed

Software Process Models and Requirements Engineering

one after another in a strict, linear sequence. In this model, the final software output is only delivered after all the phases or processes are fully completed.

The Waterfall Model is primarily used for smaller projects, as it is relatively simple and straightforward. However, it does not support frequent updates or changes, as it follows a single cycle without the flexibility for iteration.

The stages of the Waterfall Model are as follows:



2.2 Spiral Model

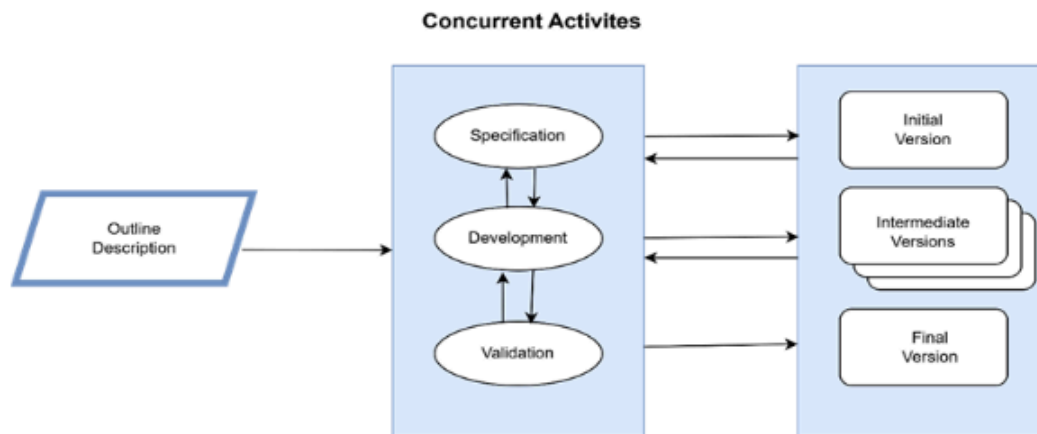
The Spiral Model is a versatile and universal software development life cycle model. Like the Iterative Model, it focuses on breaking down the development process into smaller cycles within larger SDLC phases. This approach simplifies the complexities often associated with traditional software engineering lifecycles, making it easier to manage risks and adapt to changes. However, this level of efficiency comes with a trade-off: developers may need to invest more time in tasks such as risk analysis and planning. Despite this, the Spiral Model remains one of the most effective approaches for achieving gradual product improvements, making it a valuable choice among software development life cycle models.



2.3 Incremental Model

The Incremental Model is a software development approach where requirements are divided into multiple standalone modules. Each module goes through its own set of phases, including requirements analysis, design, implementation, and testing. With every new release, additional functionality is added to the previous version, gradually building up the system. This process continues iteratively until the complete system is achieved.

The Incremental Model allows for flexibility and early delivery of partial functionality, making it easier to manage and adapt to changes throughout the development process.



2.4 Analysis of the Hybrid SDLC Model for Flipkart:

Since Flipkart is a dynamic and large-scale e-commerce platform, it requires a hybrid approach combining Incremental Development and the Spiral Model for the following reasons:

2.4.1 Why Incremental Development?

- Flipkart frequently introduces new features, such as AI-based recommendations, UPI payment options, and enhanced logistics tracking.
- By delivering updates in small, manageable increments, Flipkart ensures that new functionalities can be integrated smoothly without disrupting the existing system.
- Customer feedback is gathered and incorporated after each update, enabling continuous improvement and refinement of the platform.

- This model allows Flipkart to adapt quickly to market trends, such as flash sales or festival discounts, by deploying targeted updates in a timely manner.

2.1.2 Why Spiral Model?

- Flipkart faces complex risks, such as cybersecurity threats, payment gateway integration challenges, and scalability issues.
- The Spiral Model’s iterative risk assessment ensures that potential security vulnerabilities are identified and addressed before a feature is deployed to production.
- By conducting early-stage testing and simulations, costly failures are minimized, especially for high-risk features like fraud detection systems and AI-driven pricing algorithms.
- This approach also facilitates the seamless integration of third-party APIs with minimal disruptions to the existing system.

| Aspect | Waterfall Model | Incremental Model | Spiral Model |
|--------------------------|---|--|---|
| Development Approach | Linear, sequential process | Develops in small, incremental stages | Iterative cycles with risk analysis |
| Flexibility | Rigid | Moderate flexibility | High flexibility |
| Time-to-Market | Slow | Faster | Moderate |
| Cost Efficiency | Lower initial cost but expensive to fix errors | Moderate, spreads cost over increments | Higher cost due to extensive risk analysis |
| Scalability | Low | High | Very high |
| Suitability for Flipkart | Not ideal due to evolving requirements and frequent updates | Best for adding new features like payment integrations, AI recommendations, and UI changes | Ideal for handling security risks, scalability, and payment gateway integration |

2.5 Functional and Non-Functional Requirements for Flipkart

2.5.1 Functional Requirements

Functional requirements define the core operations and features that a system must support to meet business needs. For Flipkart, these requirements are centred on delivering a seamless e-commerce experience for users, vendors, and administrators.

1. **User Authentication & Authorization:** The system must allow users to securely register, log in, and manage their accounts. It should also implement role-based access control to differentiate between customers, sellers, and administrators.

2. **Product Management:** Sellers and administrators should have the ability to add, update, and remove products. This includes defining product categories, setting stock levels, and managing pricing.
3. **Order Processing:** The system must enable customers to add products to their cart, proceed to checkout, confirm orders, and track shipments efficiently.
4. **Payment Integration:** Secure payment gateways must be integrated to support transactions via UPI, credit/debit cards, net banking, and digital wallets. Encryption and fraud protection mechanisms should also be in place.
5. **Search & Filtering:** Customers should be able to search for products using keywords and apply advanced filters based on price, brand, category, ratings, and availability.
6. **Customer Support:** The system should include features like live chat support, an FAQ section, and clear policies for returns, refunds, and issue resolution.
7. **Personalized Recommendations:** AI-driven product suggestions should enhance user engagement by recommending items based on browsing history, preferences, and previous purchases.
8. **Multi-Vendor Support:** The platform must function as a marketplace, allowing third-party sellers to list and manage their products while tracking sales and inventory.

2.5.2 Non-Functional Requirements

Non-functional requirements define the overall system qualities that ensure efficiency, security, and usability. These attributes are critical for Flipkart's performance, reliability, and scalability.

1. **Scalability:** The system must be capable of handling millions of users, concurrent transactions, and high traffic loads, particularly during peak events like festive season sales.
2. **Security:** Robust data protection mechanisms, such as encryption, multi-factor authentication, and fraud detection, should be implemented to prevent data breaches and unauthorized access.
3. **Performance:** Flipkart must ensure fast page loading, quick response times, and optimized search results to deliver a seamless and efficient shopping experience.
4. **Availability:** The platform should guarantee a 99.99% uptime to ensure continuous operations, minimizing downtime that could lead to financial losses and customer dissatisfaction.

5. **Usability:** The user interface should be intuitive, easy to navigate, and responsive across various devices, including desktops, tablets, and smartphones.
6. **Compliance:** Flipkart must adhere to legal regulations such as GDPR for data privacy and PCI DSS for secure payment processing to ensure customer data protection.
7. **Maintainability:** The system should follow a modular architecture to simplify updates, bug fixes, and enhancements without disrupting the entire platform.
8. **Reliability:** Flipkart should incorporate failover mechanisms and redundant backups to prevent data loss and ensure quick system recovery in case of failures.

2.6 Risk and Change Management

2.6.1 Risk Management:

Flipkart faces multiple risks due to its large-scale e-commerce operations. Effective risk management is essential to minimize disruptions and maintain customer trust.

- **Security Risks:** Cyber threats, payment fraud, and data breaches are significant concerns. To address these, Flipkart implements encryption, multi-factor authentication, and regular security audits.
- **Scalability Risks:** High traffic during major sales events can lead to server overload. To mitigate this, Flipkart relies on cloud-based scaling and load balancing solutions.
- **Vendor Risks:** Unreliable third-party sellers can affect product availability and customer satisfaction. Flipkart addresses this by enforcing strict seller policies and monitoring seller performance.
- **Regulatory Risks:** Compliance with data protection laws, such as GDPR and PCI DSS, is critical to avoid legal penalties and ensure customer data security.

2.6.2 Change Management:

Consistent system updates are crucial for maintaining competitiveness and enhancing user experience. Our approach emphasizes iterative development, incorporating key principles:

- **Incremental Feature Releases:** New features, like AI recommendations or payment options, are deployed gradually for performance testing and minimal user disruption.
- **Data-Driven Improvements:** User feedback and analytics inform system enhancements and bug fixes, ensuring alignment with user needs.
- **Proactive Risk Management:** Spiral model iterations facilitate early issue detection, mitigating failure risks.

Software Process Models and Requirements Engineering

- **Efficient Development:** Agile adaptation minimizes rework and optimizes development cycles for cost and time efficiency.

2.7 Time Constraints

- **Frequent Updates Required:** Flipkart needs continuous improvements (e.g., UI changes, payment integrations) to stay competitive.
- **Peak Season Demands:** Events like Big Billion Days require rapid scalability and pre-planned updates.
- **Quick Bug Fixes & Maintenance:** Real-time issue resolution is necessary to avoid customer dissatisfaction.

2.8 Cost Constraints

- **Infrastructure Costs:** Maintaining cloud servers, databases, and content delivery networks is expensive.
- **Security Investments:** Ensuring secure transactions and data protection requires continuous financial investment.
- **Development & Maintenance Costs:** Regular updates, feature enhancements, and testing add to ongoing costs.

2.9 How SDLC Models Affect Time & Cost for Flipkart

- **Waterfall Model:** Lower initial cost but high rework expenses due to late-stage testing; slow feature releases.
- **Incremental Model:** Moderate cost spread over time; faster releases, making it cost-efficient.
- **Spiral Model:** Higher cost due to extensive risk management but reduces long-term losses from failures

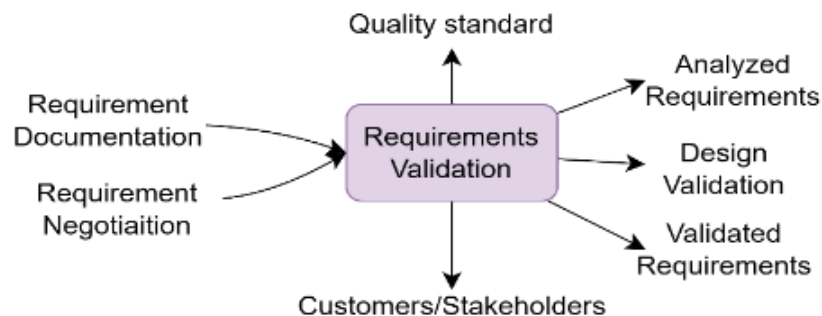
| Characteristics | Waterfall Model | Incremental Model | Spiral Model | Best Fit for Flipkart |
|-------------------|-----------------|-------------------|------------------------------|-----------------------|
| Project Structure | Rigid | Modular | Iterative with risk analysis | Incremental & Spiral |

| Risk Handling | Poor | Moderate | High | Spiral |
|------------------|--|---------------------------------------|--|----------------------|
| Time-to-Market | Long, full system developed before release | Faster, core functions deployed early | Moderate, depends on iterations | Incremental |
| Testing Approach | Done at the end, errors detected late | Continuous testing per increment | Iterative testing with risk mitigation | Incremental & Spiral |

CONCLUSION: Based on the comparison, a combination of the Incremental and Spiral models is the best choice for Flipkart. The Incremental Model enables continuous updates and faster time-to-market, which is essential for an ever-evolving e-commerce platform. Simultaneously, the Spiral Model ensures effective risk management, security, and scalability—key factors for handling large-scale financial transactions and dynamic customer demands. By integrating both models, Flipkart can maintain agility, security, and competitiveness in the market.

3. Requirements Validation

Requirement's validation is nothing but making sure that all the requirements are in agreement with the stakeholder's requests. In other words, it is about checking if the requirements are complete, consistent, and correct.



5.1 Requirements Validation Strategy

Requirements validation ensures that Flipkart's system meets business needs before full-scale development. The key strategies include:

Software Process Models and Requirements Engineering

- **Stakeholder Review:** Gathering feedback from customers, sellers, and administrators.
- **Prototyping:** Developing early UI mock-ups and interactive demos for validation.
- **Requirement Traceability Matrix (RTM):** Ensures each requirement is linked to a deliverable.
- **Automated Testing & Verification:** Running test cases to check requirement compliance.
- **User Acceptance Testing (UAT):** Conducting real-world trials with selected users.

5.2 Challenges in Requirements Validation

- **Changing Business Needs:** E-commerce trends shift rapidly, making static requirements obsolete.
- **Diverse Stakeholder Needs:** Balancing different user expectations can be challenging.
- **Scalability Considerations:** Ensuring new features won't affect performance at scale.
- **Security & Compliance:** Validating security measures against evolving cyber threats.

By combining Incremental and Spiral models, Flipkart ensures robust requirement validation while maintaining flexibility to adapt to market needs.

4.Final Conclusion

After evaluating various SDLC models—such as Waterfall, Incremental, and Spiral—it is evident that Flipkart requires a flexible, scalable, and risk-aware approach to software development. The Waterfall Model, while structured, is too rigid for a dynamic company like Flipkart, which frequently updates and adapts its platform. The Incremental Model is well-suited for delivering updates quickly and continuously improving the platform, but it may fall short in addressing risks and security concerns, which are critical for a large-scale e-commerce platform. On the other hand, the Spiral Model excels in risk assessment and security but may lack the speed needed for rapid feature deployment.

A hybrid approach combining the Incremental and Spiral Models emerges as the optimal solution. This approach allows Flipkart to leverage the speed and flexibility of the Incremental Model for continuous updates and improvements while incorporating the risk management and security strengths of the Spiral Model. This balance ensures that Flipkart can remain competitive, meet customer demands, maintain operational efficiency, and uphold security and compliance standards.

This analysis highlights the importance of selecting the right SDLC model based on business needs, technological requirements, and market dynamics. By adopting a hybrid model, Flipkart can foster innovation, enhance performance, and achieve its long-term goals. Ultimately, choosing the right development approach is key to success, and for Flipkart, a hybrid model proves to be the most effective strategy

References

1. Data <https://www.datarob.com>
2. GeeksforGeeks: <https://www.geeksforgeeks.org> .
3. Draw.io: <https://app.diagrams.net/>
4. ChatGPT: <https://chatgpt.com/>

