**PROJECT REPORT**

**ON**

**Software Development Lifecycle (SDLC) Analysis of an Online Food Delivery System (Swiggy)**

**Submitted to** NMAM Institute of Technology, Nitte  
(Off-Campus Centre, Nitte Deemed to be University, Nitte – 576110, Karnataka, India)  
On partial fulfilment of the requirements for the award of the

**Degree of Bachelor of Technology**  
**In**  
**INFORMATION SCIENCE AND ENGINEERING**

**By**  
**Chirag Rai**  
**NNM23IS037**

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

{chiragrai2005@gmail.com, nnm23is037@nmamit.in}

**Keywords:** SDLC, Software Process Models, Waterfall, Incremental, Spiral, Food Delivery System, Requirements Engineering

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| Titles | Page Numbers |
| Abstract | 3 |
| Chapters: |  |
| 1. Introduction and Overview | 4 |
| * 1. Overview of Swiggy |
| 1. Comparative analysis of SDLC models | 5 |
| 2.1 Waterfall Model |
| 2.2 Incremental Model | 6 |
| 2.3 Spiral Model | 7 |
| 1. Requirements Engineering | 8 |
| 3.1 Requirements Definition |
| 3.2 Requirements Validation Strategy |
| 3.3 Potential Challenges in Requirements Validation | 9 |
| 1. Model Selection and Analysis | 10 |
| 4.1 Functional and Non-Functional Requirements |
| 4.2 Risk and Change Management |
| 4.3 Time and Cost Constraints |
| 4.4 Model Selection |
|  |  |
| References | 11 |

**ABSTRACT**

The rapid growth of online food delivery systems has transformed the way customers interact with restaurants and delivery services. These platforms facilitate food ordering, payment processing, and real-time tracking, significantly enhancing user convenience. However, developing such a system requires careful selection of a suitable Software Development Lifecycle (SDLC) model to ensure reliability, security, and scalability.

This report examines three widely used SDLC models—Waterfall, Incremental, and Spiral—to determine their effectiveness in building an online food delivery system. Each model is analysed in terms of functional and non-functional requirements, risk management, time, and cost constraints. Their suitability is compared based on the level of flexibility they offer for handling evolving customer needs and technological advancements.

Furthermore, the report provides a structured approach to requirements engineering, including requirements validation strategies to ensure software quality. Key challenges such as changing user expectations, integration complexities, and performance optimization are discussed. This study serves as a guide for selecting the most appropriate SDLC model for food delivery platforms, ensuring a seamless user experience and efficient system performance.

**Chapter 1: Introduction and Overview**

The online food delivery industry has witnessed exponential growth in recent years, driven by advancements in technology and changing consumer preferences. Platforms like Swiggy, Uber Eats, and Zomato have become integral to modern lifestyles, offering convenience, variety, and real-time tracking. However, developing such a system requires a robust Software Development Life Cycle (SDLC) to ensure functionality, scalability, and user satisfaction.

This report explores the suitability of three SDLC models—Waterfall, Incremental, and Spiral—for building an online food delivery system, specifically Swiggy. It also delves into the requirements engineering process, highlighting functional and non-functional requirements, validation strategies, and potential challenges.

**1.1 Overview of Swiggy**

Swiggy was founded in 2014 by Sriharsha Majety, Nandan Reddy, and Rahul Jaimini in Bengaluru, India. The company was created with the vision of transforming the food delivery landscape by providing a seamless and efficient platform for customers to order food from local restaurants. With a strong focus on logistics and technology, Swiggy quickly expanded its reach to multiple cities across India and became a leading player in the online food delivery industry.

The platform offers a user-friendly interface that allows customers to browse restaurants, check menus, place orders, and track deliveries in real-time. Swiggy introduced several innovations, including cloud kitchens, express deliveries, and partnerships with grocery stores to enhance its service offerings. The introduction of Swiggy Genie for parcel delivery and Swiggy Instamart for grocery deliveries further diversified its business model.

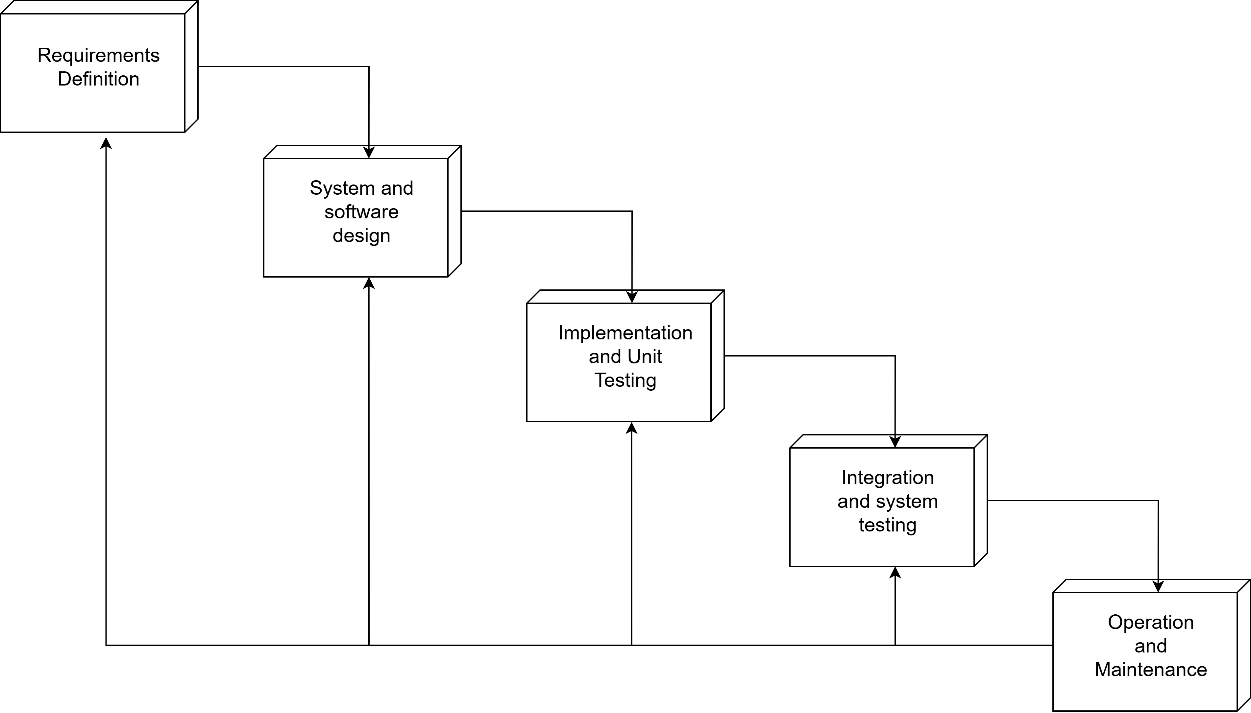
Swiggy's popularity can be attributed to its strong logistics network, AI-driven recommendations, multiple payment options, and customer-centric approach. The company continues to evolve, incorporating advanced technologies such as machine learning and data analytics to optimize delivery times, improve customer experience, and expand its services across different verticals.

**Chapter 2: Comparative Analysis of SDLC Models**

**2.1 Waterfall Model**

The Waterfall Model follows a linear sequential approach, making it ideal for projects with well-defined requirements.

* **Advantages**:
  + Clearly defined phases (Requirement Analysis, Design, Implementation, Testing, Deployment, Maintenance).
  + Documentation-driven process ensures clarity.
  + Suitable for structured development in a stable environment.
* **Disadvantages**:
  + Not adaptable to changes once the process begins.
  + Late-stage testing may lead to costly errors.
  + Poor flexibility for evolving customer requirements.

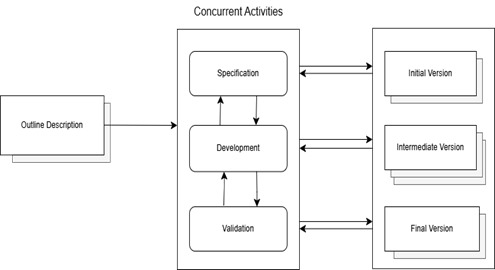


*2.1 Simple representation of a Waterfall Model*

**2.2 Incremental Model**

The Incremental Model breaks the system into smaller modules, developed and tested in iterations.

* **Advantages**:
  + Early delivery of functional components.
  + Easier risk management and bug fixes.
  + Can accommodate changing customer requirements.
* **Disadvantages**:
  + Requires careful planning and modular design.
  + Integration complexity as new modules are added.
  + High dependency on initial module success.



*2.2 Simple diagram of Incremental model*

**2.3 Spiral Model**

The Spiral Model is risk-driven and iterative, making it suitable for complex projects with evolving requirements.

* **Advantages**:
  + Emphasizes risk assessment and early error detection.
  + Allows for iterative refinement of requirements.
  + Flexible to accommodate new technologies and user needs.
* **Disadvantages**:
  + High development costs due to iterative cycles.
  + Requires skilled risk management expertise.
  + May extend project duration due to repeated cycles.



*2.3 Simple diagram of a Spiral model*

**Chapter 3: Requirements Engineering**

**3.1 Requirements Definition**

**Functional Requirements**

* User Registration and Authentication
* Restaurant Listing and Search
* Order Placement and Tracking
* Payment Gateway Integration
* Customer Support and Feedback System
* Delivery Partner Management
* Promotional Offers and Discounts

**Non-Functional Requirements**

* Scalability to handle high traffic
* High security for transactions
* Reliability and fault tolerance
* Fast response time
* User-friendly interface
* **Compliance with data protection regulations**

**3.2 Requirements Validation Strategy**

To ensure that the software meets the specified requirements, Swiggy follows a comprehensive validation strategy that includes:

* **Requirement Reviews:** Stakeholders review the requirements to verify accuracy and feasibility.
* **Prototyping:** Developing mock-ups and wireframes to validate user needs.
* **Test Cases and Scenarios:** Designing test cases to verify functional and non-functional requirements.
* **User Feedback Sessions:** Gathering input from end-users to refine the requirements further.
* **Automated Testing: Running automated scripts to check performance, security, and usability.**

**3.3 Potential Challenges in Requirements Validation**

* **Evolving User Expectations:** Customer demands change rapidly, making it difficult to finalize requirements.
* **Integration Complexity:** Coordinating multiple stakeholders, including restaurants, delivery partners, and payment gateways.
* **Performance Issues:** Ensuring system responsiveness during peak usage hours.
* **Security Concerns:** Protecting sensitive user data and payment information from cyber threats.
* **Regulatory Compliance: Adhering to local and global data protection laws.**

**Chapter 4: SDLC Model Selection and Analysis**

**4.1 Functional and Non-Functional Requirements**

* The Waterfall model ensures structured documentation of functional and non-functional requirements, but lacks flexibility.
* The Incremental model enables phased enhancements, making it more adaptable.
* The Spiral model refines requirements through iterations, ensuring continuous improvement.

**4.2 Risk and Change Management**

* The Waterfall model struggles with changes due to its rigid structure.
* The Incremental model allows gradual risk mitigation
* The Spiral model prioritizes risk assessment, making it the most effective for dynamic platforms like Swiggy.

**4.3 Time and Cost Constraints**

* **The Waterfall model is cost-efficient but lacks flexibility.**
* **The Incremental model balances cost and adaptability.**
* **The Spiral model ensures risk management but increases development time and cost.**

**4.4 Model Selection**

**In conclusion the Spiral Model may be the most effective for analysing the SDLC of a system such as Swiggy. However, if the budget is tight, then the Incremental Model may prove to be the better alternative.**

### **References**

1. **Swiggy Official Website**  
   Swiggy. (2024). Swiggy: Online Food Delivery & More. Retrieved from <https://www.swiggy.com/>
2. **Waterfall Model Explanation**  
   GeeksforGeeks. (2024). Software Engineering - Waterfall Model. Retrieved from https://www.geeksforgeeks.org/software-engineering-waterfall-model/
3. **Incremental Model Overview**  
   Tutorialspoint. (2024). Software Development Life Cycle (SDLC). Retrieved from https://www.tutorialspoint.com/software\_engineering/software\_development\_life\_cycle.htm
4. **Spiral Model Explanation**  
   IBM. (2024). Spiral Model in Software Engineering. Retrieved from https://www.ibm.com/docs/en/rational-soft-arch/9.7?topic=SS2LQC\_9.7.0/com.ibm.xtools.modeler.doc/topics/cspiral.html
5. **Swiggy Business Model and Growth**  
   YourStory. (2019). How Swiggy Became India’s Largest Food Delivery Platform. Retrieved from https://yourstory.com/2019/12/how-swiggy-became-india-largest-food-delivery-platform
6. **Requirements Engineering Best Practices**  
   ScienceDirect. (2024). Requirements Engineering in Software Development. Retrieved from <https://www.sciencedirect.com/topics/computer-science/requirements-engineering>