DrillBit

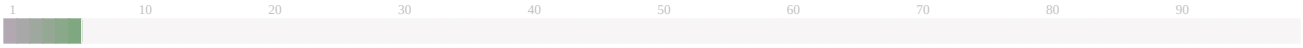
The Report is Generated by DrillBit Plagiarism Detection Software

Submission Information

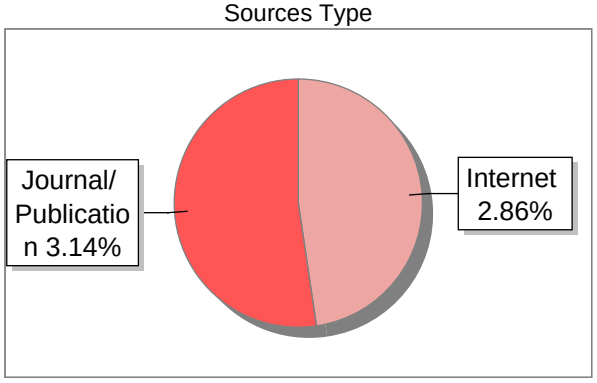
Author Name	Chirag
Title	Task
Paper/Submission ID	3331901
Submitted by	nnm23is030@nmamit.in
Submission Date	2025-02-14 13:28:20
Total Pages, Total Words	21, 2621
Document type	Project Work

Result Information

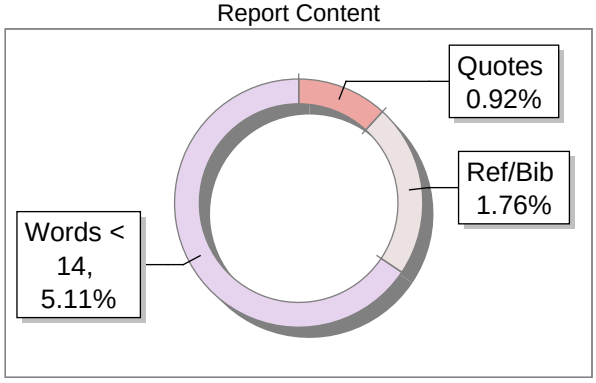
Similarity **6 %**



Sources Type



Report Content



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

DrillBit Similarity Report

6	13	A
SIMILARITY %	MATCHED SOURCES	GRADE

A-Satisfactory (0-10%)
B-Upgrade (11-40%)
C-Poor (41-60%)
D-Unacceptable (61-100%)

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	Thesis submitted to shodhganga - shodhganga.inflibnet.ac.in	1	Publication
2	ir.canterbury.ac.nz	1	Publication
3	core.ac.uk	1	Publication
4	moam.info	1	Internet Data
5	moam.info	1	Internet Data
6	docshare.tips	<1	Internet Data
7	guides.visual-paradigm.com	<1	Internet Data
8	www.volopay.com	<1	Internet Data
9	atharvacoe.ac.in	<1	Publication
10	Beam profile monitor for antiproton-nucleus annihilation cross section measureme by Todoroki-2012	<1	Publication
11	fastercapital.com	<1	Internet Data
12	Thesis Submitted to Shodhganga Repository	<1	Publication
13	IEEE 2017 IEEE 3rd International Conference on Electro-Technology f, by Oluwadamilola, K. O- 2017	<1	Publication

PROJECT REPORT

ON

Swiggy's Software Development Process: A Hybrid Approach

Submitted to

NMAM INSTITUTE OF TECHNOLOGY, NITTE

(Off-Campus Centre, Nitte Deemed to be University, Nitte - 574 110, Karnataka, India)

In partial fulfilment of the requirements for the award of the

Degree of Bachelor of Technology

In

INFORMATION SCIENCE AND ENGINEERING

By

CHIRAG

NNM24IS513

Under the guidance of

Dr. Jason Elroy Martis, Associate Professor,

Department of Information Science and Technology,

NMAM Institute of Technology. Nitte Karnataka, India

Introduction

Overview of the Assignment

When choosing process model, it is very important for the success of a project with the growing speed of software development. This report covers different software development process models and how each would work and how it would affect the importance of requirements management. In this case, we study the Incremental Development, Spiral Model and Waterfall Model to perceive their strength and weaknesses, and how suitable they are for each type of project that can be implemented. The findings are put in context with the framework of Swiggy, one of the leading food ordering and delivery platform in the world, to provide actionable insights and suggestions on how one should manage requirements in systems that are dynamic and changing.

Case Study Selection

Introduction to Swiggy

Launched in 2014, Indian meal delivery market space dominant Swiggy operates in a wide range of cities in India and is ready to spread to other countries. Users are connected with a massive variety of restaurants through the platform, making it very convenient for smooth and quick food delivery to the user's door step. These cities and regions involved with Swiggy's operations call for scalable operations and performance to support increasing user and order volumes.

Significance of the Case Study

There are several reasons why selection of Swiggy case study is important.

1. Dynamic Market Environment: With lots of competition in the food delivery market, the players are always forced to innovate and undergo change all the time, as it's the key to survive.

2. Swiggy handles multiple cities and regions, therefore it requires scalability and performance capabilities sufficient to adapt to the growing user and order volumes.

3. User Experience: As such, this platform requires the management of requirements to meet high requirements of user satisfaction and retention.

Purpose of the Study

This study aims to:

1. Compare and Contrast Process Models: Critically analyse these process models in large scale (or) in term relative to their respectively suitability for large scale software projects (generally referred to as big software projects) such as...

2. Assessing The Effect On Requirements Management: Determine how the process models affect functional and non functional requirements management.

3. This can provide practical insights: Provide actionable recommendations of how to deal with requirements in dynamic and changing systems such as Swiggy.

Importance of Software Development Lifecycle (SDLC)

Structured approach of designing high quality software is called Software Development Lifecycle (SDLC). It includes a number of phases, from planning, then requirements gathering, design, implementation, testing, deployment, and maintenance. It is at each

stage where the final product is checked to confirm it has met the specified requirements and it does function efficiently.

Research Questions

1. What is the best process model for Swiggy's dynamic environment?

2. What is the effect of different process models on the managing of functional and

non functional requirements?

3. How can the choice of this attributes validation process model be constrained? What are the core challenges and potential solutions for each process model?

Scope and Limitations

In another perspective, this report will focus on comparing Incremental Development, Spiral Model and Waterfall Model with the help of Swiggy. The analysis is thorough, but is limited to that case study and those process models. Other process models can be incorporated into such future research.

Structure of the Report

The report is organized as follows.

1. Introduction: Provides an overview of the assignment, case study selection, and purpose of the study.

2. Analysis and Comparison of Process Models: The Incremental Development Model, Spiral Model, and Waterfall Model are analysed and compared on the basis of their suitability for the company Swiggy.

3. Comparative Evaluation: Evaluates the advantages and disadvantages of each process model and their impact on project management.

4. Methods: An exploration of the application of process models in the setting of Swiggy is made with a focus on challenges encountered and their solutions.

5. The listing section presents both the research materials and supplementary reading materials used in this work

Incremental Development

Overview and Key Principles

The system development method known as Incremental Development builds the system through separate phases that successively add new capabilities to earlier versions. The method follows an iterative development model along with continuous feedback loops and it delivers services in successive increments.

Suitability for Swiggy

1. Functional and Non-Functional Requirements

Functional Requirements benefit from Incremental Development because the system enables regular new feature additions which includes restaurant partnerships

alongside enhanced screens and better order tracing mechanisms. Each added functional requirement goes into its own segment of development which ensures that the system grows step-by-step.

This modeling approach enables ongoing improvements of performance as well as scalability and user experience. The system achieves its non-functional requirements including response time and reliability alongside security measures through individual treatment within each development increment.

2. Risk and Change Management

Risk Management benefits from incremental development since systematic risks become detectable right at the beginning of every development stage. Control measures for project risk become more effective through this methodology which provides fast responses to possible issues.

The adaptive nature of Incremental Development makes it highly suitable for handling management of changes. New requirements which enter the system after the first phase can be added to following increments resulting in minimal interruptions to process operations. This approach helps system managers keep their project's integrity intact while adapting to new requirements.

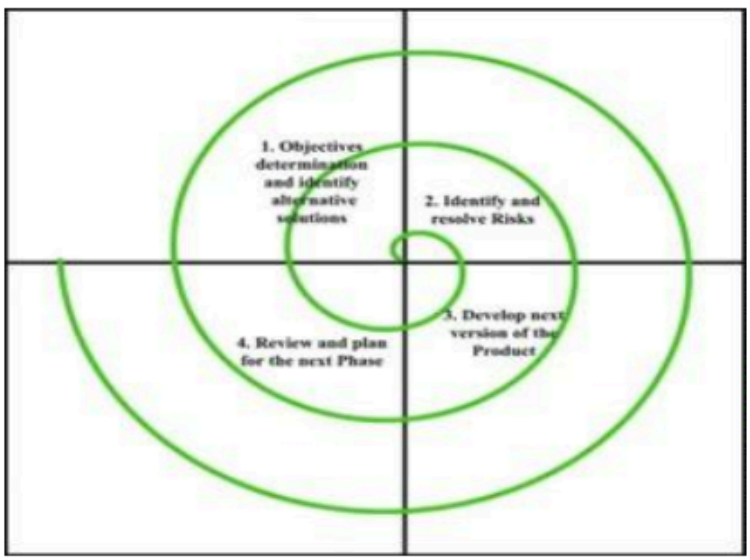
3. Time and Cost Constraints

The approach delivers an initial release of basic system functionality at an early stage so users gain prompt value from it. Each phase of development through this approach

enables system operation which leads to faster completion time.

Cost reductions take effect due to the limited scope of change since developers need to modify only the present increment. The resource allocation through this repeated process makes the development costs more efficient.

Spiral Model



Overview and Key Principles

The Spiral Model incorporates systematic Waterfall model practices through multiple development loops that work similarly to Incremental Development approaches. The approach consists of multiple loop phases which follow the

development process. A complete cycle within the Spiral Model requires separate steps for planning, risk analysis, development and evaluation.

Suitability for Swiggy

1. Functional and Non-Functional Requirements

Each Spiral Model loop enables the construction of a prototype to test and improve functional requirements. Repetition occurs through development loops to verify every functional requirement for complete documentation prior to starting actual development work.

Non-Functional Requirements are supported through an iterative approach offered by this model. All loops feature a risk analysis phase so team members can detect and prevent possible threats before they occur.

2. Risk and Change Management

The Spiral Model performs risk analysis in every development cycle which lets teams identify potential risks before they launch risk mitigation strategies. The organized approach enables developers to discover and handle possible problems at an early phase of development.

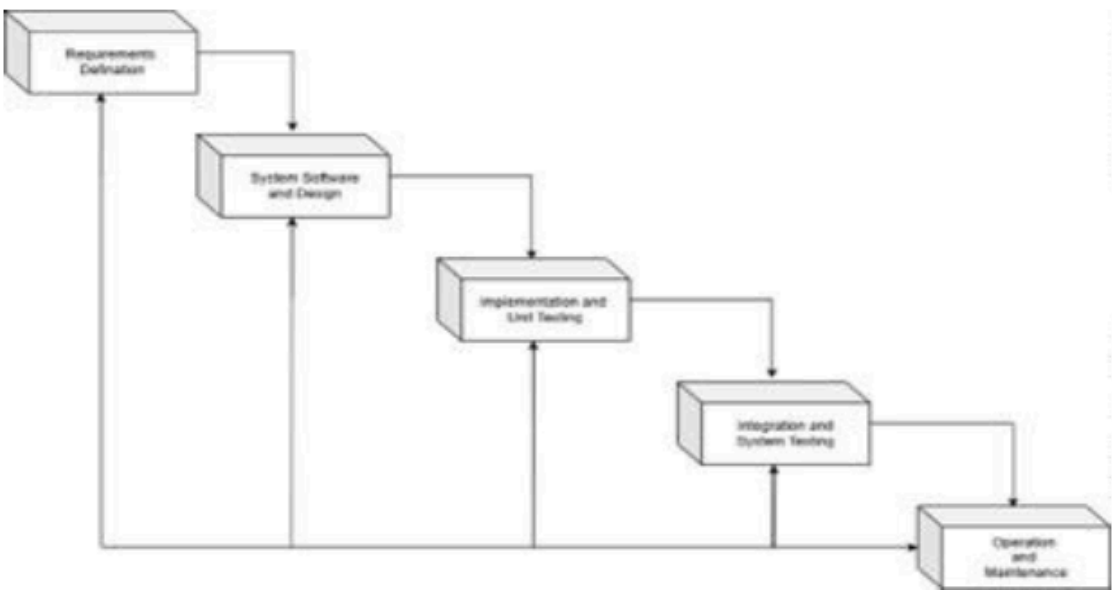
Each loop of the Spiral Model promotes simpler implementation of changes and new requirements because of its iterative organizational method. The system becomes more manageable through change since each individual loop establishes a moment to both validate and refine the system.

3. Time and Cost Constraints

Using this model developers can produce early prototypes which help execute time reduction while obtaining feedback to cut down general development periods. The systematic approach of this model enables system operation at each phase which consequently minimizes the development period.

The approach works to distribute resources effectively which lowers total development expenses. The Spiral Model proves efficient in cost management through its systematic approach to handle risks alongside changes.

Waterfall Model



Overview and Key Principles

Each stage of the Waterfall Model requires completion before starting the following stage. The model follows a linear sequence of six steps starting from planning to requirements analysis then design followed by implementation and verification along with maintenance activities.

Suitability for Swiggy

1. Functional and Non-Functional Requirements

All functional requirements receive complete definition and documentation under the Waterfall Model before starting the development process. The systematic methodology guarantees the system will satisfy all described functional requirements.

The design phase implements non-functional requirements while their final testing occurs during the last development stages. This model tests and confirms that systems fulfill all performance-based requirements and security functions and user usability standards.

2. Risk and Change Management

As the Waterfall Model completes its phases before moving on it becomes less successful at handling risk because alterations become challenging to implement after phase completion. Delay project deadlines and budget growth can occur because early risk identification is essential for the Waterfall Model to prevent major problems.

The implementation of changes using the Waterfall Model turns out to be both expensive and time-consuming because substantial rework becomes necessary. System changes become increasingly challenging to handle and maintain system integrity when using the Waterfall Model because the model lacks adaptability.

3. Time and Cost Constraints

Its linear design causes the Waterfall Model to consume considerable time for project completion. Development times become longer due to the requirement that phases need to finish entirely before starting the following phase.

High expenses will arise from changes with late discovery in the development stage. Cost overruns together with project delays become substantial problems when working with this development model.

Requirements Engineering Process

Simplified Requirements Document for Swiggy

Functional Requirements

1.User Registration and Login

Users must possess the ability to build and oversee their user profiles.

Datasecurity is achieved through secure login methods.

Multi-factor authentication for enhanced security.

1. Restaurant Listing and Search

Users must have the ability to conduct searches by various criteria such as cuisine together with location and ratings for restaurants.

Advanced search options for specific dietary requirements

2. Menu Browsing and Ordering

The system will enable users to browse through menus while making selections and initiating order placement activities.

The system needs to integrate restaurant management systems which enable real-time updates to menus.

Users will find options to modify their orders by changing ingredients during the ordering process.

3. Payment Gateway Integration

Users require secure payment methods which the system should enable through numerous payment options.

Integration with popular payment gateways (e.g., PayPal, Stripe).

Users need support for digital wallets in addition to the option for cash on delivery when making purchases.

4. Order Tracking and Status Updates

The system should display immediate updates regarding user order statuses throughout the entire process.

Users must receive alerts about order confirmation followed by delivery status notifications together with estimated delivery times.

5. User Reviews and Ratings

Users must have access to both review and rating functions of their experience.

Users should have the capability to post their reviews and ratings through social media connections.

6. Push Notifications for Order Updates

To enhance user experience the application should deliver push notifications whenever users require important updates and additional information.

Customizable notification settings for users.

Non-Functional Requirements

1. Performance

The system needs to tolerate heavy visitor counts while avoiding extended wait times during operation.

The system must provide short delays to enable time-sensitive information updates and delivery of notifications in real-time.

2. Scalability

The system needs to have capabilities to expand its capacity as the number of users and restaurants increases.

The system requires horizontal scaling as a method to manage higher system loads.

3. Usability

Users require a user interface which enables intuitive navigation through the system.

The system provides accessibility features that accommodate users with disabilities.

4. Security

The system needs data protection strategies that maintain private information security and confidentiality for both users and restaurant operators.

Compliance with data protection regulations (e.g., GDPR).

5. Availability

The system needs to operate continuously throughout 24 hours with very minimal operational interruptions.

High availability architecture with redundant systems.

Strategy for Requirements Validation

1. Prototyping

The team should build prototype systems for every development increment to collect stakeholder verification of requirements.

System performance becomes more accurate through the receipt of early feedback from stakeholders.

2. User Testing

The team should schedule user testing events to collect information regarding the system's usability together with its functional capability.

The system needs to fulfill user requirements while offering convenient use to all users.

3. Beta Testing

The beta version deployment to select users becomes a vital step to uncover remaining issues followed by their repair.

The system achieves reliability and robustness by receiving feedback from its deployment in operational environments.

4. Automated Testing

The implementation of automated tests will confirm the system achieves its entire set of functional along with non-functional requirements.

The system function will maintain reliability through error reduction and consistent performance.

Potential Challenges in Requirements Validation

1. Changing Requirements

Users with shifting expectations might result in considerable work and delay during the project.

A continuous feedback system enables organizations to handle changing requirements throughout development operations.

2. User Feedback

The practice of user feedback collection along with its implementation demands significant time investment from the organization.

The use of proper feedback processes alongside efficient communication systems helps to simplify the process.

3. Testing Environment

Gaining an exact testing environment which duplicates the production environment requires significant effort.

Cloud-based testing platforms that include simulation tools help resolve this issue.

4. Resource Constraints

The available resources sometimes create limits that reduce the possible scope of testing and validation work.

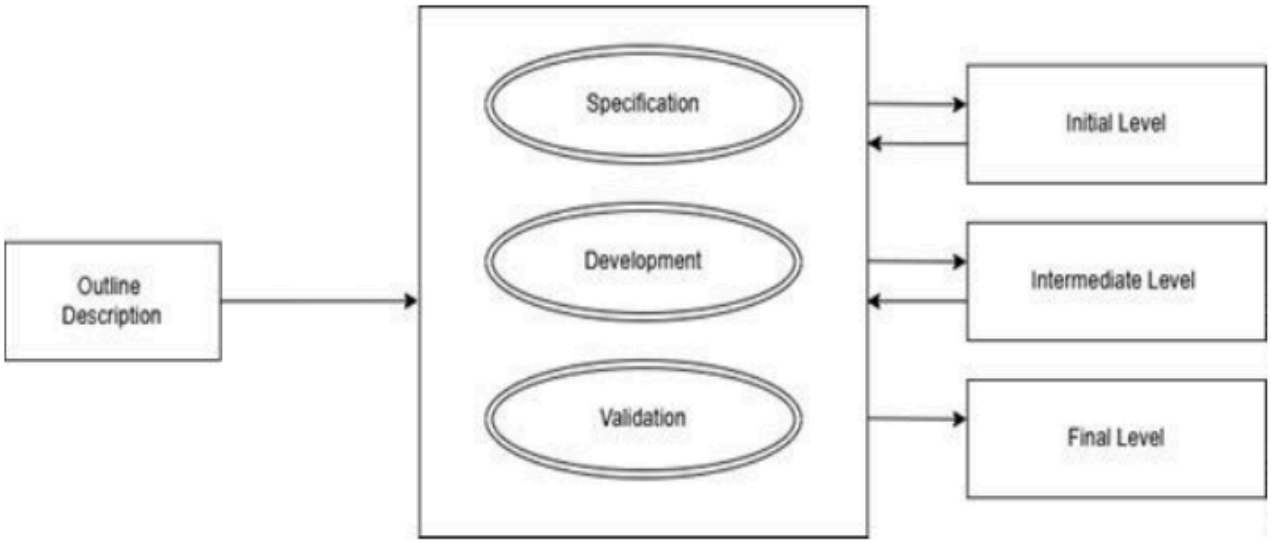
The challenge can be reduced by giving priority to resource allocation in an efficient manner.

Conclusion

The selected process model determines the level of success that software development projects achieve. Swiggy selects Incremental Development alongside the Spiral Model because these methods provide the required flexibility and adaptability to address dynamic requirements through positive user experiences. The foundational base Swiggy received from its Waterfall Model development exists but this approach becomes less applicable for sustaining evolutionary growth and performance enhancement.

Swiggy can adapt to its changing food delivery market through Agile methodologies combined with the strengths of various process models. Researchers should investigate mix-model software development approaches for other business sectors to boost current operational methods.

Practicing Incremental Development



Swiggy employs Incremental Development to consistently roll out new features and services. For instance, recent releases have included additional restaurant categories, enhanced search filters, and improved order tracking. This methodology enables Swiggy to swiftly adapt to evolving user requirements and market conditions.

Practicing Incremental Development

Swiggy employs Incremental Development to consistently roll out new features and services. For instance, recent releases have included additional restaurant categories, enhanced search filters, and improved order tracking. This

methodology enables Swiggy to swiftly adapt to evolving user requirements and market conditions.

Challenges and Strategic Solutions

Addressing Dynamic Requirements

Swiggy encounters the challenge of managing dynamic requirements due to the competitive landscape of the food delivery industry. To mitigate this, Swiggy employs continuous feedback mechanisms and agile methodologies to ensure that new features and services effectively address user needs.

Ensuring Scalability and Performance

Given its operations across various cities and regions, Swiggy requires a scalable and high-performing infrastructure. To tackle this, Swiggy employs cloud-based solutions and horizontal scaling techniques to accommodate growing user and order volumes.

Promoting Security and Usability

Swiggy prioritizes user experience to uphold high levels of user satisfaction and retention. To achieve this, Swiggy implements advanced security protocols and user-friendly accessibility features, ensuring both data protection and a seamless interface.

References

"Definition of Incremental Development: Benefits and Challenges"

"A Comparative Study of Zomato and Swiggy"

"Swiggy Business Model & Growth Strategy - GrowthX Deep dive"

This paper provides a complete guide about the Spiral Model in Software Engineering.

"Waterfall Model in Software Engineering: A Comprehensive Guide"