DrillBit

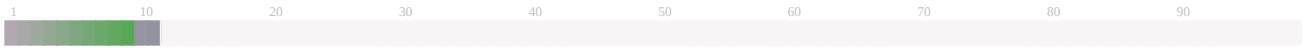
The Report is Generated by DrillBit Plagiarism Detection Software

Submission Information

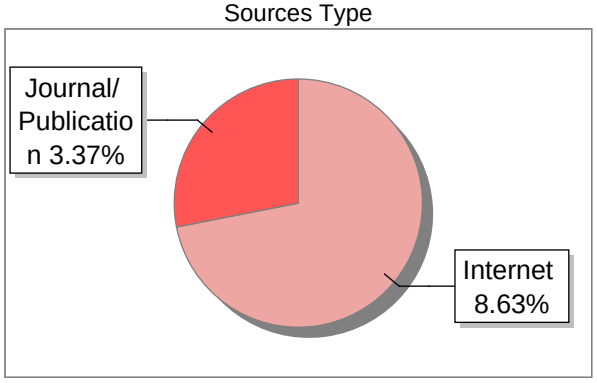
Author Name	Dhanraj
Title	Task
Paper/Submission ID	3335291
Submitted by	nnm23is030@nmamit.in
Submission Date	2025-02-16 08:19:21
Total Pages, Total Words	10, 1870
Document type	Project Work

Result Information

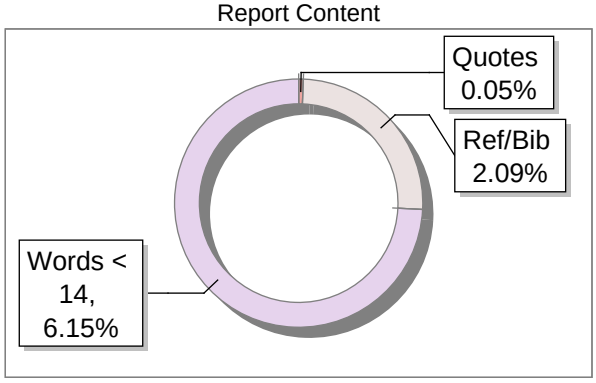
Similarity **12 %**



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

12	15	B
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	learnerbits.com	2	Internet Data
2	inoxoft.com	2	Internet Data
3	Auditing within service-oriented architecture the Dutch experience by Guah-2013	1	Publication
4	www.readbag.com	1	Internet Data
5	link.springer.com	1	Internet Data
6	Crystal Engineering with the New Linker Tolanedisulfonic Acid (H by Zitzer-2015	1	Publication
7	timesofindia.indiatimes.com	1	Internet Data
8	www.sec.gov	1	Internet Data
9	pdf4pro.com	<1	Internet Data
10	redcol.minciencias.gov.co	<1	Publication
11	Thesis submitted to shodhganga - shodhganga.inflibnet.ac.in	<1	Publication
12	www.bgsvijnatham.com	<1	Internet Data
13	www.irdindia.in	<1	Publication

14	www.linkedin.com	<1	Internet Data
15	www.simplilearn.com	<1	Internet Data

Software Development Lifecycle (SDLC) Analysis of

BigBasket

A Comparative Study of Different Models in Relation to BigBasket Software Development

Dhanaraj Y Shetty

Nitte Mahalinga Adyantaya Memorial Institute of Technology

dhanarajyshetty91@gmail.com, nnm24is505@nmamit.in

https://github.com/dhanarajyshetty/sdlc_bigbasket_analysis/edit/main/README.md

Keywords:SDLC, BigBasket, AWS, Integration, Testing, Scalability, Requirements validation

Abstract:

It discusses the software development life cycle (SDLC) models specific to Big Basket, a popular online grocery delivery store. A comparative study of SDLC methodologies and an intro to requirements engineering. The intent of the study is to provide an understanding about which SDLC model should be selected for large scale e commerce platforms emphasizing on Waterfall, Incremental Development and Spiral Model (Big Basket) objectives. The report also looks at problems and strategies in testing requirements and deployment issues faced by Big Basket. This paper establishes the findings based on extensive research, industry best practices and from Big Basket's technology stack. The report can be a significant source for software engineers, architects and researchers involved in the cross of related to SDLC/requirements How e-commerce platforms deal with requirements. A Conclusion and Contributions follow this paper. The report also discusses challenges and strategies involved in requirements validation and software deployment at BigBasket. The findings in this document are based on the extensive research, industry best practices, and insights from BigBasket's technology stack. This report serves as a valuable resource for software engineers, architects, and researchers interested in the intersection of SDLC and requirements software engineering methodologies in large-scale e-commerce platforms. This paper is followed by a conclusion and considerations.

Publication:

This paper is hosted on a GitHub repository, along with the material used for preparing this resea

Table of Contents

1: Introduction

Big Basket is one of the leading online grocery delivery services, leveraging technology to streamline operations and enhance customer experience. This report examines how different Software Development Life Cycle models apply to Big Basket's platform and explores the best practices used for developing scalable, secure, and high-performing e-commerce applications. Additionally, we discuss how Software Development Life Cycle models impact Big Basket's ability to innovate in a competitive e-commerce industry, ensuring that their technology stack supports high availability and optimal performance.

2: Overview of Big Basket

Big Basket is a company delivering online grocery delivery to customers in India which served millions of consumers across various cities. The services offered by the company and its user-friendly site help customers shop for products effectively as well, order them easily with quick delivery. After all, its very nature centres upon an expansive supply chain network that places tens of thousands of products within reach in every facility at a given moment.

Operating via several fulfilment models, including warehouses, dark stores and tie-ups with local Kirana's that can give it quick commerce capabilities as well traditional scheduled deliveries. Offering a wide range of products, from fresh fruits and vegetables or groceries to household items as well as gourmet articles. The answer to that, of course is Big Basket which as ever focused on technology, logistics and customer delight doing what it does best continues its innovation streak in the e-commerce — grocery delivery market!

2.1 System Overview

Big Basket system architecture consists of various interdependent components, including:

- Order management
- Payment processing
- Inventory management
- Last-mile delivery tracking

2.2 Technologies Used

Backend: Java and Python (Spring Boot, Django)

Frontend: React, Angular

Database: MySQL, MongoDB

Cloud Services: AWS (EC2, S3, Lambda, DynamoDB)

AI & Data Analytics: Machine learning models for recommendation engines, demand forecasting, and automated customer support

3: Comparative Analysis of SDLC (Software Development Life Cycle) Models

3.1 Waterfall Model

Waterfall Model is old style. It is linear and sequential. Each phase has to be completed before the next phase can commence. This fits very well for projects that have well-defined and stable requirements. It is a linear and sequential approach to software development that consists of several phases. It must be completed in a specific order. The model ensures rigorous planning, structured testing, and compliance, making it ideal for highly regulated components of BigBasket's architecture. However, its rigidity makes it less suitable for frequent feature updates, which require adaptability. Despite this limitation, the Waterfall Model remains relevant for foundational system development, ensuring long-term reliability in Big Basket's e-commerce operations.

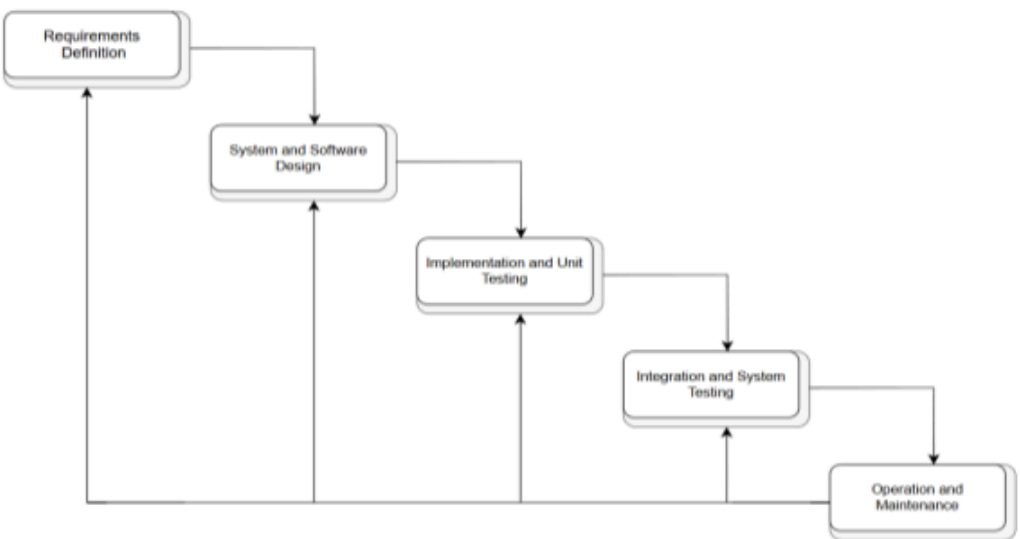


Fig. 1. Waterfall Model

Advantages:

- Structured and well-documented process
- Suitable for projects with fixed requirements
- Easier to track progress

Implementation at Big Basket:

Requirement Analysis: Initial business objectives and system requirements are clearly defined before development begins.

Design Phase: The entire system, including microservices and cloud infrastructure, is designed in one go.

Implementation: The complete system is built at once, integrating order management, inventory tracking, and payment processing.

Testing: Comprehensive testing is conducted after full implementation, including functional, performance, and security testing.

Limitations: This model lacks flexibility for evolving requirements and may not suit Big Basket's need for frequent updates.

3.2 Incremental Development Model

Incremental model also known as the successive version model is a process of software development where requirements divided into multiple standalone modules of the software development cycle. In an e-commerce platform like BigBasket, the Incremental Model plays a crucial role in maintaining agility and responding to market demands. The first increment may focus on core functionalities such as user registration, product catalog management, and basic order processing.

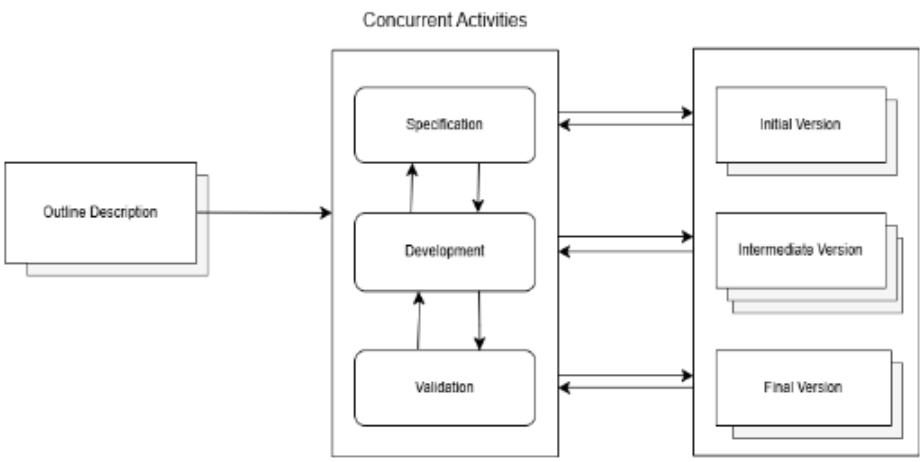


Fig. 2. Incremental Model

Advantages:

- Quick release of functional components
- Adaptable to customer feedback
- Lower risk in software development

Implementation at Big Basket:

First Increment: Core functionalities like user registration, product listing, and search capabilities are developed and deployed.

Subsequent Increments: Additional features such as personalized recommendations, secure payments, and delivery tracking are gradually integrated.

Feedback-Driven Development: Customer reviews and usage data influence further improvements.

Continuous Integration and Deployment (CI/CD): Regular updates improve system performance and user experience.

3.3 Spiral Model

The Spiral Model is a sophisticated software development lifecycle (SDLC) model that integrates key elements from both the Waterfall and Iterative models while placing a strong emphasis on risk management. This model is particularly suited for complex and large-scale projects where uncertainties and evolving requirements are common.

The Spiral Model follows a cyclical process, consisting of four major phases in each iteration: Planning, Risk Analysis, Engineering, and Evaluation. Each cycle, or "spiral," represents a progressive refinement of the software, allowing for continuous improvements and risk mitigation at every stage.

Fig. 3. Spiral Model

Advantages:

- Systematic risk assessment
- Iterative refinement of software components
- Strong focus on scalability and security

Implementation at BigBasket:

Risk Identification: Each phase is begins with an assessment of security, scalability, and performance risks.

Prototype Development: Initial versions of modules like payment gateways and AI-based recommendations are tested before full implementation.

Iterative Enhancement: Systematic improvements are made based on risk analysis and customer feedback.

Final Deployment: The refined and tested system is deployed after multiple cycles of risk-based assessments.

3.4 Suitability for BigBasket

BigBasket operates in a fast-paced e-commerce environment where customer demands, market trends, and technology evolve rapidly. To support its growing business needs, a hybrid SDLC approach that integrates aspects of both the Incremental and Spiral Models proves to be the most effective. The Incremental Model allows BigBasket to develop and release features in smaller, manageable phases, ensuring continuous improvements while gathering customer feedback. Meanwhile, the Spiral Model's emphasis on risk assessment at each phase helps BigBasket mitigate potential security vulnerabilities, scalability challenges, and integration issues before full deployment.

Additionally, implementing DevOps methodologies alongside a hybrid SDLC approach further streamlines development, testing, and deployment processes. DevOps enables continuous integration (CI) and continuous deployment (CD), allowing BigBasket to automate software releases, enhance system reliability, and accelerate time-to-market for new features. This is particularly crucial for optimizing order processing, improving recommendation algorithms, and enhancing mobile and web application performance

Summary of Comparison

SDLC Model	Strengths	Challenges for BigBasket
Waterfall	Well-structured, good for stable requirements	Lacks flexibility for evolving market needs
Incremental	Quick updates, user feedback integration	Requires effective change management
Spiral	Risk-focused, scalable development	Complex and resource-intensive

4: Requirements Engineering for BigBasket

4.1 Functional Requirements

- User authentication and account management
- Product search, filtering, and recommendation system
- Order placement, modification, and tracking
- Secure payment gateway integration
- Customer support via chatbots and the human agents

4.2 Non-Functional Requirements

- Scalability to handle high traffic
- High availability and fault tolerance
- Data security and compliance with regulations
- Fast response time for better user experience

4.3 Requirements Validation Strategy

- Regular stakeholder meetings
- Prototyping and iterative feedback
- Automated testing frameworks for the performance validation

4.4 Challenges in Requirements Validation

- Rapidly changing the customer expectations
- Integration complexity with the third-party vendors
- Ensuring compliance with the multiple regulatory standards

5: Case Study on BigBasket

Problem Statement:

As BigBasket expanded its operations across multiple cities, it encountered significant challenges in scaling its infrastructure while maintaining a seamless user experience. The growing customer base and increasing order volumes required a robust, scalable, and high-performance system to handle demand surges, especially during peak hours and festive seasons.

One major issue was inventory management inefficiencies, which led to stockouts, delays in order fulfillment, and discrepancies between displayed and actual product availability. Additionally, logistical bottlenecks affected delivery timelines, particularly in high-density urban areas where traffic and last-mile delivery posed challenges.

Ensuring data security and fraud prevention was another critical concern, as handling large volumes of sensitive customer data, including payment details and delivery addresses, required robust security measures.

To overcome these challenges, BigBasket needed to adopt a scalable and resilient technology architecture, optimize its supply chain operations, and implement advanced automation and AI-driven solutions to maintain its competitive edge in the e-commerce industry.

Solution:

- Adoption of Microservices Architecture: Improved system flexibility and modularity
- Cloud-based Infrastructure: Enhanced performance and reliability
- Machine Learning Integration: Provided personalized recommendations and demand forecasting

Outcome:

- Increased system uptime to 99.9%
- Enhanced customer satisfaction through personalized shopping experiences
- Reduced order processing time by 40%

6: Conclusion

BigBasket's adoption of a hybrid SDLC model underscores the significance of flexibility, scalability, and risk management in the ever-evolving landscape of e-commerce software development. By integrating elements from various methodologies, the company is able to streamline development processes, accelerate time-to-market, and efficiently address the dynamic demands of online retail. This approach not only enhances system performance and security but also ensures seamless user experiences by enabling rapid adaptation to changing customer preferences and market trends. Furthermore, the hybrid model allows for proactive risk mitigation, reducing potential disruptions and ensuring business continuity. Ultimately, this strategic adoption positions BigBasket as a resilient and innovative player in the competitive e-commerce industry.

7.References

Big basket official website - https://www.bigbasket.com/

AWS Case Study: BigBasket. (n.d.). *How BigBasket Uses AWS to Scale Operations*. Retrieved from https://aws.amazon.com/solutions/case-studies/bigbasket/

TOI Tech Desk. (2022). *BigBasket's Expansion Strategy and AI-driven Personalization*. Times of India. Retrieved from https://timesofindia.indiatimes.com/