PROJECT REPORT

On
Software Development Life Cycle(SDLC) Analysis of Amazon

A comparative study of different models in relation to Amazon System Development

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

Chaithanya N Kamath
NNM23IS035

Under the guidance of

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

INDEX

Github Repository: https://github.com/chaithanyakamath/Amazon-SDLC

1.Introduction

1.1. What is SDLC?
SDLC stands for Software Development Lifecycle. It is a process used by software development teams and other IT professionals to design, develop, test and deploy high quality software. Basically it provides a framework for planning, creating and maintaining software systems by ensuring that they meet user requirements and are delivered on time and within budget to the customer.

1.2 Why Is SDLC so Important?
SDLC is a step by step process which guides the development of Software, which starts from initial planning and analysis to deployment and its maintenance. SDLC is very crucial in Software Engineering because it provides us a structured and systematic way to develop high quality software.

Some of its Importance:
- It ensures us to build a High Quality Software
- It generally reduces the risks in the process
- Cost and Time Efficiency is improved.
- It gives us a clear cut Roadmap of the journey
- It improves the Customer Satisfaction

1.3 How does It work?
Working is divided into several Stages, since software goes through several stages of fixing bugs and correction as developers add new versions with new features added in them.
General steps involved are:

- **Planing**: In this process we define Scope and Objective of the Software.We identify the resources by conducting some required studies.

- **Requirements Analysis:** In this process they gather and document the business requirements and also identify the functional and non-functional requirements of the Software.
- **Design :** Here they create design specifications and prototypes and also choose the technology stack.
- **Development:** Writing and implementation of code by following the guidelines and coding standards begins at this stage.
- **Testing:** Various tests are conducted to identify and fix the defects for quality assurance.
- **Deployment:** After successful testing they release the software into the production environment and keep monitoring for post deployment issues.
- **Maintenance:** After deploying the software they need to provide ongoing support and updates to ensure its performance. They also need to implement new features based on the user requirements.

Planning

↓

Requirements
Analysis

↓

Design

↓

Development

↓

Testing

↓

Deployment

↓

Maintainance
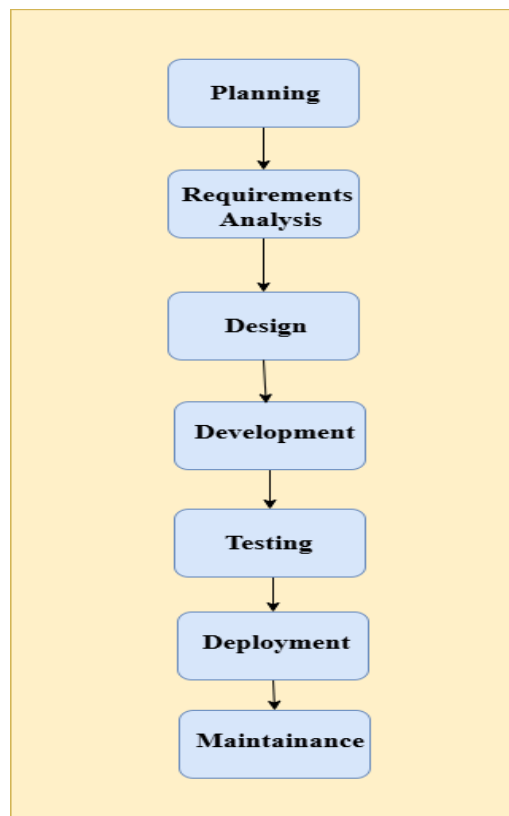
1.4 SDLC Models

There exists different SDLC models because Software development projects vary in complexity, requirements and risk factors. We can work with different SDLC models as per our software requirements.

Some of the popular models are:

- Waterfall model
- Incremental Development Model
- Iterative Model
- Agile Model
- Spiral Model
- V Model

Here, we will be mainly focusing on Waterfall Model and Incremental development Model

2. Analysis of Some Models.

2.1 Waterfall Model:

Waterfall Model is a linear and sequential approach to Software development where each phase needs to be completed in order to enter its next phase, overlapping between two phases is not possible here. It is considered to be one of the easiest and simplest SDLC models.
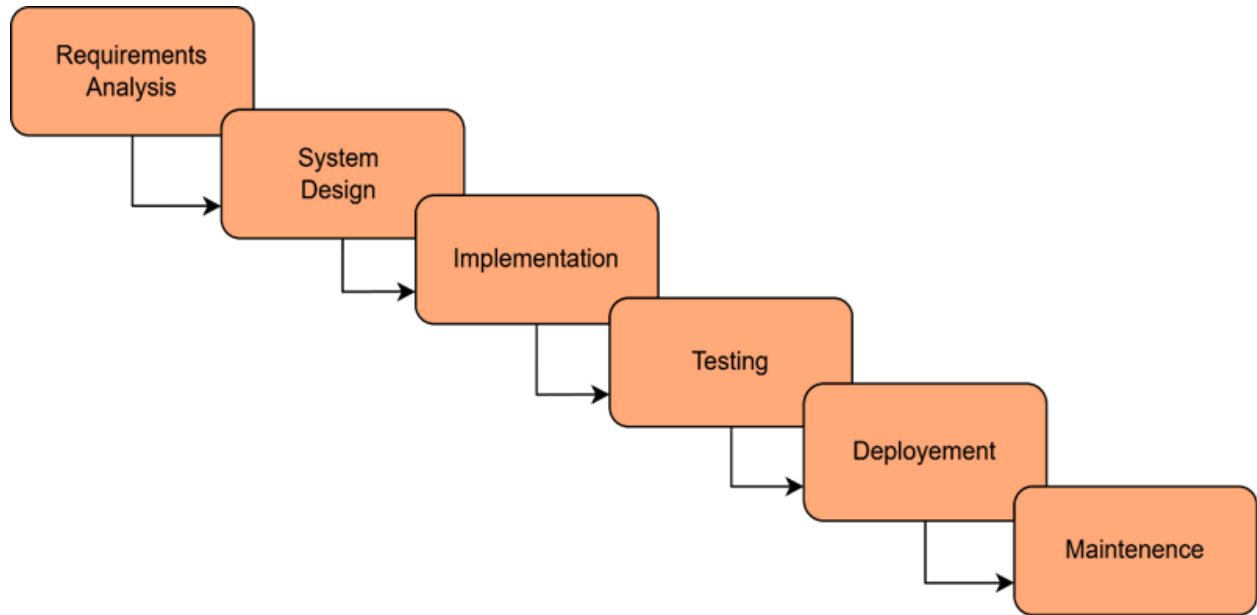
2.2 Incremental Development Model:

It is a Software development approach where the systems are built and delivered in increments. Each increment adds new features and builds upon the previous existing feature until the complete system is built or developed.
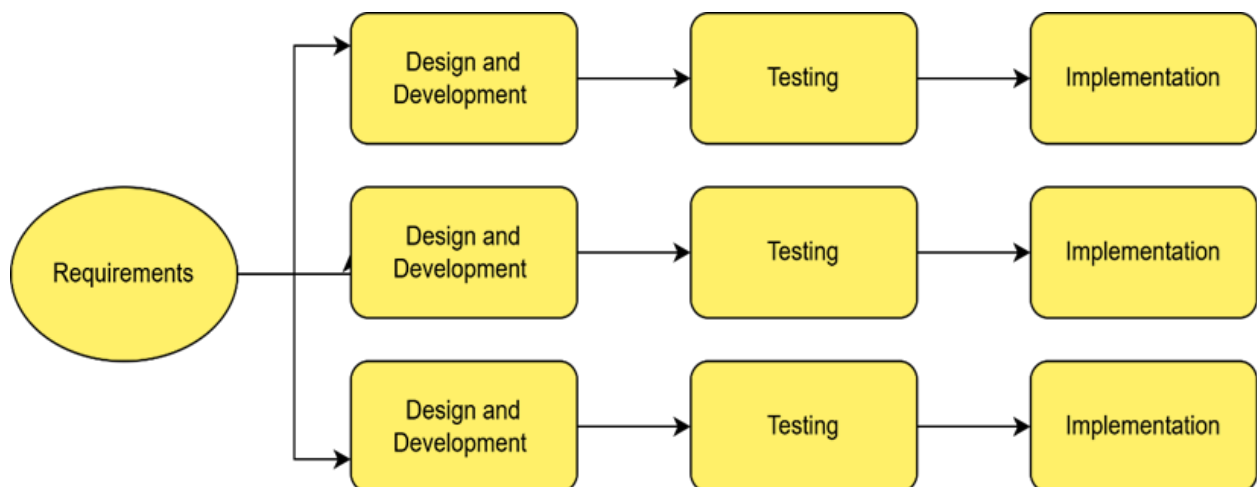

2.3 Iterative Model:

In this model, software is built and improved in repeated cycles(Iterations) hence it is called the 'Iterative Model'. Each Iteration produces a working version of the software, allowing for incremental improvements or developments.

2.4 Spiral Model:

It is a risk-driven SDLC that combines Iterative development with some elements of Waterfall Model. It is generally ideal for large, complex and high-risk projects. It is called so because it has multiple loops, where each loop represents a phase in the development process.
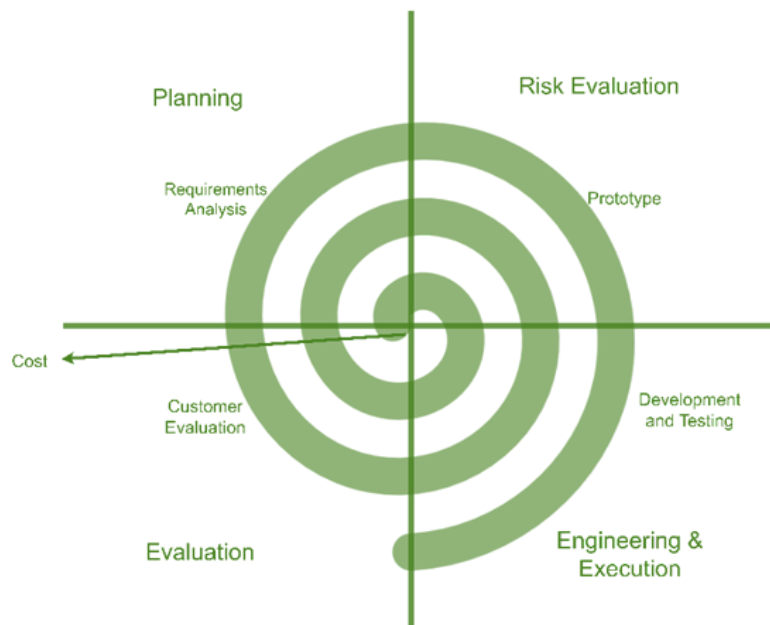
2.5 How to choose a perfect Model

Selection of a good model plays an important role in the Software development Process which generally depends on various factors.

Steps involved in choosing a perfect model are:
1. Analyze your Project Requirements
2. Access project Complexity and Size
3. Identify the level of risks involved in the project

4. Determine the involvement of customers

5. Evaluate time-to-market

6. Consider the availability of budget

7. Testing and quality assurance

Some of the most important factors the determine the suitability are

• **Function and Non-Functional Requirements.**

• **Risk and change management.**

• **Time and cost constraints.**

2.6 Functional Requirements

Functional Requirements define the specific behaviours, features and functionalities a software or system must perform.

Main features include:

• Define the actions the system must perform.

• Describe the interactions of the system with users and others.

• Ensure that all required business processes are covered.

In case of <u>E-commerce Platform</u> i.e **Amazon**, that provides Online Shopping, Order Management and delivery services has the following functional requirements:

User Management:

- User Register and login using email, phone or social media
- User updating their personal details, passwords and addresses.
- Role based access must be given.

Product Management:

- Sellers must list their products with proper descriptions, prices and images.
- Product categorization and tagging must be supported by the system.

- Real time based product availability must be implemented.

Search and Filtering:

- Searching products by using keywords must be implemented.
- Filters must be available for various topics such as price range, brand, rating etc
- Personal recommendation as per history must be provided.

Shopping Cart and Wishlist:

- Adding/ removing items from the cart must be easy for the user.
- Details like total price, delivery date and discounts must be displayed on the cart .

Order Management:

- Order must be trackable with real time status update.
- Multiple payment options to place the order just be provided.

Payment Processing:

- Secured transaction with enabled inscription must be there.
- Generate invoices and payment receipts to customers.

Customer Reviewing and Rating:

- Rating and reviewing the purchased products must be easy for customers.
- Reviews should be filtered to avoid spam reviews.
- Verified purchase tags should be displayed.

2.7 Non-Functional Requirements

Non-Functional Requirements are the requirements that defines how well a system performs which includes its performance, stability and reliability.
Main features include,

• Defining the system quality attributes.

• Affects the UI/UX but does not specify how it does it.

• Ensures the system meets the business goals

In case of <u>E-commerce Platform</u> i.e **Amazon**, that provides Online Shopping, Order Management and delivery services has the following non-functional requirements:

Performance Requirements:

- Website must load at least within 3 seconds
- Must support millions of concurrent users.
- Quick payment process.

Availability Requirements:

- System should have 99.99% uptime.
- Implementation of failure mechanisms to ensure its continous operation during its failures.
- Should implement distributed architecture.

Maintainability Requirements:

- Must allow zero downtime updates.
- Modular architecture systems.
- Logging and monitoring for detecting and resolving bugs.

2.8 Risk and Change Management

2.8.1 Risk Management

Risk Management is a systematic approach of identifying, assessing and correcting risks throughout the SDLC lifecycle to ensure the success of the software. Effective risk management always ensures that chances of failures, delays will be minimized.

Some of the major steps involved in Risk Management are:

- **Risk Identification:** Identifying the risks that might create an impact on the project i.e Technical risks, Operational risks, security risks etc
- **Risk Analysis:** This can be done by qualitative and quantitative approach.
- **Risk Mitigation and Planning:** Development of strategies to reduce or eliminate the existing risks.
- **Risk Monitoring and Controlling:** Continuous monitoring identified risks and also detect new ones using SDLC.

In E-commerce Platform like Amazon there will be different types of risks can be seen, some of them are:

- **Technical Risks** - Software bugs and vulnerabilities
- **Security Risks -** Cyberattacks, fraud activities
- **Operational Risks -** Logistics failures, supply chain disruptions
- **Financial Risks -** High operating costs, revenue loss

2.8.2 Change Management

Change management involves planning, implementing and monitoring changes in a controlled manner to ensure that there will be minimal disruption and also continuous improvement.

Amazon's Change Management Framework:

1. **Identification and Planning:**
    - Identifying Scope and impact of change

- Conduct a chain impact Analysis -
- Develop a clear Plan.

2. **Stakeholder Engagement and Communication:**

- Engage key stakeholders.
- Provide clear communication.
- Train the teams

3. **Implementation and Deployment:**

- Used for phase/Iterative deployment
- Depends on automated tools

4. **Monitoring and Feedback:**

- Continuous monitoring of system performance, user feedback
- Use real-time data and dashboards

5. **Evaluation and Continuous Improvement:**

- Conduct post-implementation reviews.
- Promote a culture of continuous improvement.

Examples of Change Management at Amazon:
1. Adoption of One-Day Prime delivery
2. Implementation of machine learning
3. COVID-19 adaption

Risk and Change Management are considered to be critical components in Software development to ensure project's success, maintenance and its quality.

2.9 Time and Cost Constraints:

Time and Cost constraints are considered to be essential factors which influence its operations, projects and overall business strategies. Amazon as a giant E-commerce company must carefully manage time and cost constraints in order to maintain its efficiency, innovation and its profitability. We can say that time and cost constraints make an effect on the Scope of the Company.
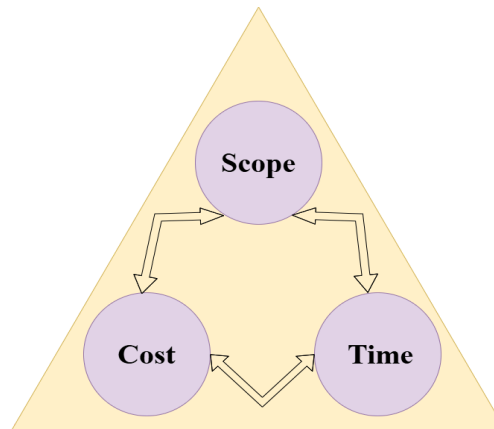
2.9.1 Time constraints:

For Amazon, Time plays a crucial role in maintaining the edge with other competitors in the market. It operates in a fast-paced and High speed environment.

- Same day/ One-day delivery commitment always imposes strict time constraints on logistics and supply chain management.
- During Prime day or Black friday sales, Amazon should prepare in advance to manage high traffic and order volumes within the given timeframes.
- Cybersecurity or System outrages require immediate resolution to minimize downtime and other possible losses.

2.9.2 Cost constraints

A company to be profitable they must do cost controls despite having huge resources. Major Cost distribution of Amazon are:

- Investments in Automation(Robots, AI)
- Maintaining and expanding AWS infrastructure.
- High Customer acquisition costs due to increasing competition.
- Offering competitive prices and free shipping for prime members.
- Adapting to new regulations.

How Amazon balances Time and Cost constraints

- Reducing Manual tasks which speeds up operation by reducing costs.

- Supporting high traffic cost efficiency.
- Reducing delivery time and logistics costs.

2.10 Choosing a Model:

Among Waterfall Model, incremental development Model and Spiral Model we need to look at various properties before selecting/choosing a best model for Amazon. They keep adding new features, constantly changing, handling high volumes of users and managing various tasks together.

For selecting a particular model, first we need to make comparisons between the available models on the basis of Risk Management, Scalability and Flexibility, Handling Complexity and Development.

Conclusion:

By going through the comparison table below, we can conclude that the Spiral Model is the best Model among all three models to build an Amazon company. This Model is obtained by comparing some of the characteristics of each model and then determining the best model which suits it.

Characteristic	Waterfall Model	Incremental Model	Spiral Model	Best among 3 Models
Risk Management	Higher Risks, since issues/ bugs are found at the end	Little lower Risks, since testing happens frequently	Less Risks, as testing in done during each cycle	Spiral Model
Scalability	Poor Scalability	Moderate Scalabity	High Scalability	Spiral Model
Flexibility	Less Flexible	Modetarly Flexible	Higly Flexible	Spiral Model
Development Speed	Slowest	Moderate to Fast	Variable	Incremental Model

3 Requirements Validation:

Requirements Validation here ensure that the requirements for new features, products or services which align with the business goals, customer requirements and technical feasibility.This process is considered to be very crucial to avoid costly errors, improve customer satisfaction and also to maintain Amazon's market leadership.

Amazon follows certain steps in Requirement Validation Process:

a) **Requirement Gathering and Documentation:**
- Requirements are collaborated by business teams, product managers , technical teams and other stakeholders.
- These requirements are documented in detail.
- Customer feedback, market research, competitive analysis and internal analysis are some of the sources.

b) **Validation Techniques:**
Validation of gathered requirements plays an important role for Amazon, so they employ multiple techniques to ensure that they are correct. Some of the techniques are:
- Peer Reviews and Walkthroughs
- Prototyping
- Test case Design
- Customer feedback
- Feasibility and risk analysis

c) **Approval and Sign Off:**

Once the gathered requirements pass validation, then they are approved by key stakeholders. This always ensures alignment between business, engineering and customer experience teams.
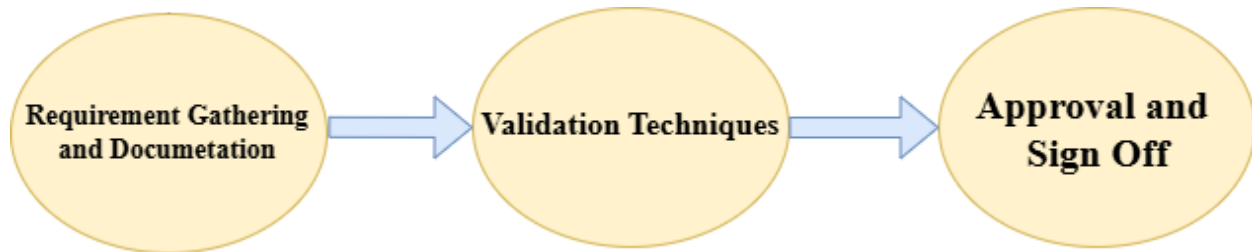
Some of the Examples of Requirements Validation in Amazon:

1) **Prime Delivery Service:**
   - **Requirement:** Offer One day delivery for Prime members.
   - **Validation:** Conduct operational feasibility tests -> Analyse logistics data -> Run customer surveys and pilot programs.

2) **AWS Service Rollout:**
   - **Requirement:** Launch a new AWS Service for scalable data storage.
   - **Validation:** Technical Validation -> Security Assessment -> Customer Feedback.

4.CONCLUSION:

This report is a brief on how Amazon uses software development life cycle Models such as *Waterfall Model*, *Incremental Development Model* and *Spiral Model* for its benefit. The selection of the SDLC played very important role in the development of a large-scale application such as Amazon

Waterfall Model is best for predictable and well defined Projects Incremental development model provides a flexible and efficient approach to Software development by breaking huge projects into smaller and manageable modules whereas Spiral model combines iterative development with a strong focus on risk analysis and management.

We can conclude that among the **Waterfall**, **Incremental**, and **Spiral** models, the **Spiral Model** is the best fit for Amazon due to its **risk management focus, iterative development, and adaptability to complex, large-scale systems.**

5.References:

1. Boehm, B. W. (1988). A Spiral Model of Software Development and Enhancement. *ACM SIGSOFT Software Engineering Notes*, 11(4), 14–24. https://dl.acm.org/doi/10.1145/12944.12948
    - This paper introduces the Spiral Model, focusing on Iterative development and risk management.

2. Sommerville, I. (2016). Software Engineering (10th ed.). Pearson Education.
    - Covers different software development methodologies.

3. Royce, W. W. (1970). Managing the Development of Large Software Systems. *Proceedings of IEEE WESCON*, 1–9.
    - Introduced the Waterfall Model.

4. Pressman, R. S. (2014). Software Engineering: A Practitioner's Approach (8th ed.). McGraw-Hill.
    - Provides guidances on software development practices.

5. Amazon. (2025). Amazon's Approach to Software Development and Innovation. *Amazon Annual Report*. Retrieved from https://www.amazon.com
    - Official document explaining Amazon's Strategies

6. IEEE. (2014). IEEE Standard for System and Software Verification and Validation. IEEE Std 1012-2012.
    - Defines best practices for Software verification and validation.