**Software Development Lifecycle (SDLC) Analysis of Amazon**

***A comparative study of different models in relation to Amazon’s software development***

**Bhagya P Shetty**

**NNM23IS032**

|  |
| --- |
| **Keywords:** SDLC, Amazon, AWS, Integration, Testing, Scalability, Requirements validation. |
| **Abstract:** |
| Software development is an evolving discipline requiring structured approaches for building scalable, secure, and efficient systems. This report explores the Software Development Life Cycle (SDLC) models applicable to Amazon, a globally known e-commerce system. It provides a comparative analysis of SDLC methodologies, an overview of requirements engineering.  The study aims to offer insights into the selection of an appropriate SDLC model for large-scale cloud-based platform, emphasizing the waterfall, incremental development and spiral model approaches. The report also discusses challenges and strategies involved in requirements validation and software deployment at Amazon.  The findings in this document are based on extensive research, industry best practices, and insights from Amazon websites. I hope this report serves as a valuable resource for software engineers, architects, and researchers interested in the intersection of SDLC and requirements engineering methodologies and large-scale streaming platforms. This paper is followed by a conclusion and few considerations. |
| **Publishing:**  This paper was submitted to *Dr. Jason Elroy Martis, Associate Professor, Department of Information Science and Technology, NMAM Institute of Technology. Nitte Karnataka, India.*  This paper is also hosted on a GitHub repository, along with the material used for preparing this research. The link to the GitHub Repository is given in the endnote. |

**Table of Contents**

1. **Introduction 3**
2. **Analysis of SDLC Models 4-11**
   * 2.1 Incremental Development Model 4-6
   * 2.2 Spiral Model 6-9
   * 2.3 Waterfall Model 9-11
3. **Requirements Engineering Process 12-16**
   * 3.1 Functional Requirements 12-13
   * 3.2 Non-functional Requirements 13-15
   * 3.3 Requirements Validation Strategy 15-16
4. **Conclusion 17**
5. **References 18**

**1.INTRODUCTION**

The Software Development Life Cycle (SDLC) is a well-structured process that is pivotal for software developers and engineers in designing, developing, testing, and maintaining software applications. SDLC is essential for maximum efficiency and quality of the requisite system. A company like Amazon, which does everything on a grand scale in developing everything from the e-commerce platform to Amazon Web Services (AWS).

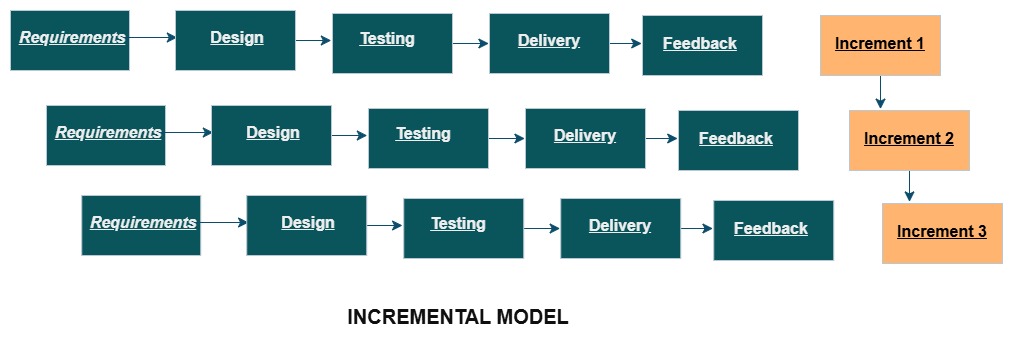
Therefore, the design of SDLC by the Amazon is greatly influenced by its scalability, performance, and rapid innovation product needs. I have chosen Amazon as the topic because it is one of the largest tech firms globally, running an incredibly intricate ecosystem that encompasses not only e-commerce but cloud services (AWS), etc.

That's an overview of the reasons for the paper on the Amazon SDLC aspect, thus enabling readers' understanding of software design, development, testing, and maintenance at Amazon. In effect, this report would show how Amazon implements SDLC methodologies to meet the business goals and remain competitive in the fast-moving business landscape of e-commerce. It could also explore providing real-world insight into problems.

**2.Analysis of SDLC models**

**2.1 Incremental Development Model**

According to Ian Sommerville, this approach adds the activities of specification, development, and validation. The system is developed as a series of types (increments), with each type adding functionality to the previous type.

  
1. Requirement Gathering: In Amazon, requirement gathering would involve user registration and login where customer create an account to use the app and customers search for the product based on some categories.

2.System Design: The design for the first increment focuses on user registration and basic user interface in amazon is mainly designed for displaying the product categories.  
3.Development & Testing: In amazon, user registration and login is tested to ensure that users can securely sign up or log in and manage their profiles. If not the development team works on building the same feature.  
4.Incremental Delivery: The first increment id delivered to the users as usable version of the app with registering. Although amazon app is not fully developed, it still provides real value and functionality to the user.

5.Feedback: In amazon, after the first increment is released, user feed back is collected this includes that users may find certain features which is difficult to navigate.

6.Next Increment: Based on the feedback from the users, the development team plans the next iteration is planned like new features such as shopping cart functionality.

* **Functional Requirements:**

The functional requirements may be assigned to smaller increments in the Incremental Development model. Amazon goes through identifying core features and building upon the release as smaller increments.  
Each increment introduces a part of the functionality, and based on feedbacks from the other increments, Amazon makes adjustments. For example, the Prime video might release basic streaming features first, then later add advanced features like modified content.

* **Non-Functional Requirements:**Non-functional requirements get incremented. Amazon ensures that the performance and scalability are tested and improved at every stage. Performance metrics could be collected on the first deliverable of a service, and optimization could occur based on real-world usage in subsequent releases.
* **Risk Management:**

Incremental Model is continuous with respect to risk management, as risks can be identified with each increment. In this way, since Amazon releases its features in smaller increments, it is able to address issues like performance degradation and security concerns much more rapidly.

* **Change Management:**

Change management is highly flexible in this model because new changes have been introduced with increments. Based on customer feedback, Amazon updates AWS services, prime video with new features.

* **Time Constraints:**

The fundamental functionality can be released earlier with this model. Amazon can put a product in the market quickly. Each increment requires some time to release.

* **Cost Constraints:**

The costs in the incremental model are quite predictable compared to the spiral model. This helps Amazon in prioritizing cost-effective deliveries.

**2.2 Spiral Model**

The spiral model is software development life cycle model that combines some aspects of both iterative development and risk management. In this case, the process will be repeated in series of spirals.



1.Planning phase: This phase of the Amazon Shopping App is stupefied to define the features and requirements for the app's next iteration. For instance, Amazon may wish to introduce a new feature such as voice search; hence development needs to plan accordingly.  
2.Risk analysis phase: Assessing whether there is a possibility of server crashes during high traffic events, such as Black Friday sales. To ascertain whether the new feature has effectively displaced the other app.  
3.Engineering phase: In amazon, improving the screen layout for easier navigation or integrating personalized recommendations. Engineers here mainly work on integrating new features such as real time notifications.  
4.Evaluation phase: In amazon after undergoing the engineering phase the new feature comes straight and tests it with real users and collects users' feedback through app reviews.  
  
5. Next spiral: The process of continuing to the next cycle begins with rectifying bugs, modifying features, and if needed, creating additions based on the user's will and requirements. Amazon will, after reviewing the voice search feature, please insert it to handle more complex queries.

* **Functional Requirements:**

The functional requirements are incorporated into the explanatory iterative cycles in the spiral model. Amazon also defines some of its features like order placement, order tracking, and providing recommendations.

* **Non-Functional Requirements:**

These features are also functional requirements as they would go through at least one iteration. With the feedback, Amazon could improve on which performance goals they would like to focus on.

* **Risk Management:**

Here, the spiral model focuses on risk analysis and management at each stage. Amazon may face security issues in the process and, therefore, be

able to develop strategies. In this regard, risk management at Amazon allows the team to test conditions, identify issues, and then solve them.

* **Change Management:**

In the spiral model, the change management is adaptable, as one can make changes between cycles. Amazon will deliver a feature that one user wants.

* **Time Constraints:**

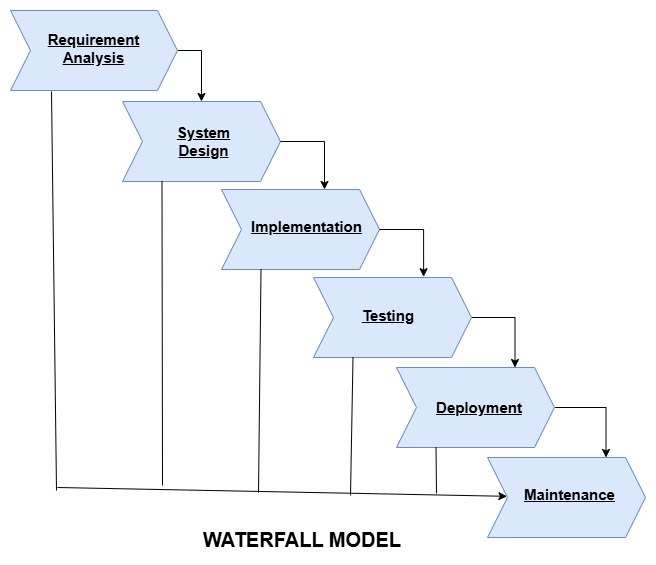
The Spiral model takes out faster releases but would also delay because of risk, development, and testing. Amazon manages to release promptly.

* **Cost Constraints:**

We know the cost is incurred at a higher rate because of so many iterations and so many risks. Amazon incurs higher operational costs due to so many iterations, which can be solved by early releases and development.

**2.3 Waterfall Model**

The sequential design process where development is seen as going steadily downwards like a waterfall. The execution of this phase is mainly one after the other.



1.Requirements Analysis: This is the phase where an Amazon shopper creates accounts, signs in to an account, and manages a profile. The shopper can search for products by category and add items to their cart.  
2.System Design: The design would require the Amazon backend system to perform various operations including inventory management, order processing, recommendations, storage of user data, and storage of product information.  
3.Implementation: In this stage, Amazon covers user authentication, product  
search, order processing, payment processing. It entails creating the login  
screen, product search screen, etc.  
4.Integration and Testing: The Amazon development team checks to see if the cart is updated every time a product is added and that the payment system processes transactions correctly.  
5.Deployment: First, the Amazon app is built for mobile platforms and then deployed to the app stores.  
6.Maintenance: Amazon incorporates features on customer feedback like voice search, and improves the existing ones. Amazon's customer support team handles user’s questions and solves complaints.

* **Functional Requirements:**

In the waterfall model, initially all functional requirements get collected and used to design and develop a system. They include order management and payment processing systems for Amazon.

* **Non-Functional Requirements:**

Similar to functional requirements, early definitions are assigned for nonfunctional requirements. Some examples of nonfunctional requirements included for Amazon are scalability, security, and availability.

* **Risk Management:**

Because the Waterfall method is inflexible to changes, any risks that are identifiable in the early stages can be addressed in the planning phase.

* **Change Management:**

Changing the goals is extremely difficult in the Waterfall Model as a change made in any step can influence the whole project process. The later the changes made in Amazon are from the original design stage, the longer the delays are likely to be.

* **Time Constraints:**

The Waterfall model is very time-bound because every stage is required to be completed before the next stage can begin. In Amazon, time constraints can delay projects when risks are identified later on.

* **Cost Constraints:**

Introducing changes at a late stage leads to high costs in the Waterfall model. Similar things can apply elsewhere, such as at Amazon, where testing and development costs are higher.

**3. Requirements Engineering Process**

**3.1 Functional Requirements:**1. **User Authentication:**

* Registration: The user is to open an account for entering required personal information.
* Login/Logout: Users must log in and out during the use.
* Password Reset Function: The user must reset their password for security purposes.

2. **Product Management:**

* Listing a Product: Sellers must collate the details of the product and upload it.
* Product Search: Customers should search for an easily accessible means for selection.
* Product Recommendations: The system has to record preferences as well as give recommendations to shoppers.

3. **Shopping Cart:**

* Addition and Removal: A customer has to put in their shopping cart that they want and take out things they do not care for.
* View Cart: Customers should be able to review their carts even before confirming any orders.

4. **Payment and Order Management:**

* Order Review: Before purchasing a product, the customer has to go through the review procedure.
* Payment Integration: The system is to accept any and all forms of payment methods.
* Order Confirmation: The customer has to get a response about their order in confirmation after they have placed it.
* Order Tracking: Customers should be able to track their orders.

1. **Customer Support:**

* Live Chat: Customer interaction with the support team takes place via live chat.
* Frequently Asked Questions: Another aspect of doubt-clearance about the products is this one.

7.**Reviews and Ratings:**

* Reviews of Products: Customers can post product reviews at will.
* Ratings of Products: Customers can give ratings for a product with stars.

8.**Seller Dashboard**

* **Order Management**: They should manage their orders.
* **Inventory Management**: They should check if a product is in stock.

9.**Notifications**

* **Email and SMS Alerts**: Messages should be sent to the customers and sellers regarding their order.
  1. **Non-Functional Requirements:**

1. **Performance**

* **Response Time**: A measure must be added into the system for response time.
* **Load Handling**: The system must cater for very high user loads with thousands of users.

1. **Scalability**

* **Horizontal Scalability**: While Internal scalability means scaling within one machine, External scaling means graduating to a bigger machine.

1. **Security**

* **Data Encryption**: It's also important that sensitive data will be encrypted.
* **Secure Payment**: That payment is secured should be defined by other common standards of practice.
* **Account Protection**: Accurate tax computation must be ensured in the system according to regional guidelines.

1. **Availability**

While such provisions for recovery from any disaster should be assured by all means, internal scalabilities substantiate that some parameters will show abruptness or variation before maintenance after service were settled.

1. **Usability**

* **User-Friendly Interface**: The system must have clean interface for customers and sellers.
* **Mobile Accessibility**: The system need to be fully responsive

**6. Compatibility**

* **Multi-format Compatibility:** They should see that their platform is compatible with all major browsers.
* **OS Compatibility**: The application should run across all prominent operating systems.

**7. Maintainability**

* **Modular Architecture**: The architecture must be such that it allows certain other modules to be written as new modules as the application grows, thus making it comparatively more maintainable.
* **Logging and Monitoring**: Application systems must provide logging and monitoring functions to indicate condition changes through time.

**8.Legal Compliance**

* **Data Privacy**: Complying with Data Privacy laws is thus the height of mortal policing.
* **Tax Compliance**: Efforts and advances toward better tax calculations should be ensured in all provinces, catering for regional laws.
  1. **Requirements Validation Strategy:**
* **Set Clear Validation Objectives:** Validate that requirements are in accordance with organizational goals.
* **Involve the Stakeholders at the Beginning and Throughout:** Ensure all key stakeholders are incorporated.
* **Conduct Gap Analysis:** A clear understanding and analysis of any requirement lacking or incomplete is needed.
* **Test Planning and Execution:** Define validation criteria and develop comprehensive test cases based on requirements.
* **Change Management:** Changes to requirements must be taken through due diligence.

**Potential Challenges:**

* **Ambiguous Requirements:** They may not be clarify and hence show the misalignment between the stakeholder and the development team.
* **Scope Creep:**  Stakeholders introduce new requirements during the validation face, resulting in scope creep.
* **Testing Difficulties:** The difficulty of validating requirements meets real-world usage thereby challenging.
* **Integration Complexity:** Amazon systems have to integrate with multiple external and many internal services thereby would prove challenging for validating requirements.
* **Market Dynamics:** Technology changes requirements less relevant.

**4.Conclusion**

The Amazon software development life cycle (SDLC), therefore, is a very involved and a very vigorous process that internationalizes the commitment of the drive for efficiency, scalability, and growth. Amazon adheres to the agile and iterative development approach whereby most of the work is subject to an ongoing improvement process to ensure that its systems-from the e-commerce platform and onto the AWS services- maintain high standards of performance, security, and user satisfaction.

The company, being heavily involved in e-commerce and cloud computing, has scalability and performance integrated from the very design phase. Heavy investment is done with regard to what is termed as nonfunctional requirements such as load handling, latency, and availability, so Amazon systems' ability to scale seamlessly as user demand rises are always guaranteed. Amazon has always set the trend in the technical world on how deliverables embody scalable, reliable applications with the user in mind.

**5.References**

1. Software-Engineering-9th-Edition-by-Ian-Sommerville.pdf -book

2. [Development and Test on Amazon Web Services - AWS Whitepaper](https://docs.aws.amazon.com/pdfs/whitepapers/latest/development-and-test-on-aws/development-and-test-on-aws.pdf) - pdf

3. [Prove It or Lose It: The Product Validation Process | Amazon Small Business Academy](https://www.smallbusiness.amazon/articles/the-product-validation-process) -website

GitHub Account: