Green University of Bangladesh

Department of Computer Science and Engineering (CSE)
Semester: (Fall, Year: 2024), B.Sc. in CSE (Day)

SDLC model selection

Exprement Name: SDLC model selection for Quiz & Earn: A Dual-Mode Quiz App with Real-Time Rewards and Competitive Leaderboard

Course Title: Integrated Design Project I
Course Code: CSE-324 , Section: 213-D1

Students Details

Name	ID
Arman Hossain	221002624
Jannatul Ferdous	221902002
Afnan Khan Shopnil	221002570

Lab Date: 14 Oct 2024
Submission Date: 21-Oct-2024
Course Teacher's Name: Rusmita Halim Chaity

[For teachers use only: Don't write anything inside this box]

Lab Report Status	
Marks:	**Signature:**
Comments:	**Date:**

1 Objectives

- To learn about the Software Development Life Cycle (SDLC) and its stages.
- To learn the different SDLC models and their characteristics.
- To understand the importance of SDLC model for app development.
- To learn how to tailor SDLC processes to meet the specific needs of the project.
- To learn the significance of user feedback and iterative improvements.
- To learn about risk management and mitigation in the SDLC
- To understand how to incorporate testing and quality assurance into the SDLC
- To learn how to efficiently manage resources, time, and budget in software development

2 Problem Analysis

The current landscape of quiz applications reveals that many focus primarily on entertainment, lacking meaningful reward systems and a strong educational foundation. These apps often limit users to online quizzes without providing a comprehensive, competitive, and rewarding experience. Additionally, most fail to create a platform that effectively combines education, competition, and tangible rewards, leaving users with an experience that is enjoyable but ultimately financially unrewarding.

3 Methodology

3.1 Stage 1: Planning and Requirement Analysis

Requirement analysis is the most critical and fundamental stage for the development of the *Quiz & Earn* app. Senior members of the development team, along with inputs from key stakeholders such as the customer, market analysts, sales department, and domain experts, perform this analysis. This phase involves gathering all the necessary information to understand the project scope and requirements for both the dual-mode quiz feature (competitive and learning modes) and the real-time rewards system.

In addition, a feasibility study is conducted in three key areas:

- **Economic Feasibility**: To evaluate whether the development costs and operational expenses align with the expected return on investment.
- **Operational Feasibility**: To assess the viability of the app in terms of user engagement, user base expansion, and the app's sustainability over time.

- **Technical Feasibility**: To determine if the required technology and tools (real-time databases, reward system integrations, leaderboard functionalities) are available and sufficient to implement the app with minimal risks.

Risk identification and quality assurance planning are also critical components of this stage. The risks, such as potential downtime or security concerns around reward systems, are identified, and mitigation strategies are formulated.
The outcome of this stage is a well-defined project approach that minimizes risks and maximizes the chances of success in developing the *Quiz & Earn* app.

3.2 Stage 2: Defining Requirements

After the requirement analysis is complete, the next step is to clearly define and document the product requirements. This is done through a *Software Requirement Specification (SRS)* document, which outlines all the necessary features and functionalities that the app will need to deliver. The SRS for the *Quiz & Earn* app will include:

- **Quiz Modes**: Details about the two modes—learning quizzes and competitive quizzes.
- **Real-Time Rewards**: Specifications on how the reward system will work, ensuring real-time accuracy and security for users.
- **Leaderboard**: Functional requirements on how the leaderboard will rank users, update in real-time, and display to competitors.
- **User Authentication**: Requirements for user sign-up, login, and security.
- **Accessibility and Usability**: Ensuring the app is user-friendly and accessible across different devices and platforms.

Once the SRS document is completed, it is reviewed and approved by the customer or market analysts. This document serves as the foundation for the subsequent stages in the development process.

3.3 Stage 3: Designing the Product Architecture

The *SRS* document serves as the reference for the product's architecture. In this stage, the product architecture for *Quiz & Earn* is designed. Based on the requirements outlined in the SRS, the development team proposes multiple design approaches, which are documented in the *Design Document Specification (DDS)*. Each design approach is evaluated based on:

- **Risk Assessment**: Considering the risks associated with various design choices, such as data security, real-time synchronization, and server load management.
- **Product Robustness**: Ensuring that the app can handle high traffic, especially during competitive quizzes.

- **Design Modularity**: Breaking down the app into modules such as the quiz engine, rewards system, leaderboard, and user management, so each component can be developed and maintained independently.
- **Budget and Time Constraints**: Ensuring that the chosen design aligns with the project's financial and timeline requirements.

The best design approach is selected based on these parameters. The selected design approach will clearly define all the architectural components, including:

- The **quiz engine** that handles the logic and presentation of quiz questions in both modes.
- The **real-time reward system** that manages and distributes rewards efficiently.
- The **leaderboard module**, which updates user scores and ranks dynamically.

The internal design of each module, including how they interact with each other and external services (such as payment gateways for rewards or external APIs for leaderboard updates), is detailed in the *DDS*. The "Quiz & Earn" project introduces a mobile application designed for the Android platform, offering an interactive quiz experience with real-time rewards. The app will be developed using Java for the frontend and PHP for backend operations, ensuring a robust and scalable architecture. It features a user-friendly interface enhanced with Lottie animations and CardView layouts, providing an engaging user experience. Additionally, the app integrates a secure payment system to facilitate cash rewards, ensuring both functionality and security for users.

3.4 Stage 4: Building or Developing the Product

In this stage, the actual development of the *Quiz & Earn* app begins. The coding is done according to the Design Document Specification (DDS). Developers follow coding standards and use tools like compilers, interpreters, and debuggers. Languages such as Java and PHP are used to build the quiz engine, rewards system, and leaderboard.

3.5 Stage 5: Testing the Product

Testing is integrated into every phase of development but focuses heavily here. In this stage, the app is tested for bugs, performance issues, and adherence to the Software Requirement Specification (SRS). Issues are fixed and retested until the app meets the required quality standards.

3.6 Stage 6: Deployment in the Market and Maintenance

Once the app passes testing, it is deployed to the market, potentially in stages for user acceptance testing (UAT). The app is released to a limited audience, feedback is collected, and necessary enhancements are made. After full deployment, ongoing maintenance is performed to address any issues and ensure smooth operation for users.

4 SDLC Models

There are various Software Development Life Cycle (SDLC) models followed in the industry for app development. Some of the most important and popular models are:

- Waterfall Model
- Iterative Model
- Spiral Model [1]
- V-Model
- Agile Model
- Prototype Model

Below is a brief overview of each model and its relevance to the "Quiz & Earn" app development:

4.1 Waterfall Model

The Waterfall Model follows a linear, sequential design process. Each phase cascades into the next, and you only move to the next phase once the previous one is completed. This model works best when the project requirements are well understood and unlikely to change. However, given that the "Quiz & Earn" app may need to adapt to user feedback and evolving market needs, the rigid nature of the Waterfall Model makes it less suitable for our project.

4.2 Iterative Model

The Iterative Model starts with a basic implementation of a portion of the software and enhances it through multiple iterations until the full system is developed. It is useful when the project scope is large and the software needs to be developed in manageable chunks. For "Quiz & Earn," this could be useful, but the complexity of features like real-time rewards and the leaderboard might require a more flexible approach to address rapid changes.

4.3 V-Model

The V-Model follows a parallel process where each development phase has a corresponding testing phase. While this model emphasizes validation and verification at each step, it is more suitable for smaller projects with well-defined requirements. Given the dynamic nature of "Quiz & Earn," where new features and updates may be frequent, this model might limit flexibility.

4.4 Spiral Model

The Spiral Model combines iterative development with risk analysis. It is ideal for large projects where frequent releases and prototyping are necessary, and risk management is a priority. The real-time rewards and leaderboard features in "Quiz & Earn" do introduce some risk in terms of system reliability and security, but the Spiral Model may be too heavyweight for a mobile app project like ours.

4.5 Agile Model

The Agile Model focuses on iterative development with short cycles (sprints), emphasizing flexibility, customer feedback, and incremental improvement. This model allows for adaptive planning and continuous delivery, making it highly suitable for projects where requirements may evolve. Given that "Quiz & Earn" will need regular updates based on user feedback, and features such as quizzes, rewards, and leaderboards will need to be refined over time, Agile provides the necessary flexibility and speed.

4.6 Prototyping Model

The Prototyping Model is used to create a working model of the product to gather early feedback from users. This model is ideal for projects where the user experience is key, and the final product may evolve after initial feedback. While valuable for gaining insights into user preferences for "Quiz & Earn," it lacks the broader development framework needed for continuous improvement.

5 Implementation in A Dual-Mode Quiz & Earn App

In this section, we evaluate various Software Development Life Cycle (SDLC) models to determine the most suitable one for our project. The models were assessed based on their compatibility with our requirements, yielding the following scores:

- Waterfall Model: 16
- V-Model: 22
- Iterative Model: 16
- Spiral Model: 17
- Agile Model: 24
- Prototyping Model: 08

Summary of Results

The Agile model scored the highest (24), making it the most aligned with our project's needs. Its flexibility and emphasis on iterative development allow for regular feedback and adjustments, ideal for creating an interactive quiz application.

While the V-Model (22) is a strong alternative due to its structured approach to testing, Agile's adaptability to changing requirements positions it as the preferred choice.

Conclusion: Why Agile is Best for Quiz & Earn

The Agile model is the best choice for "Quiz & Earn" due to:

- **Flexibility:** Supports frequent iterations, essential for real-time rewards and user experience enhancements.
- **User-Centric Approach:** Facilitates constant feedback to refine app interactions.
- **Adaptability:** Allows smooth integration of updates, particularly for the leaderboard and reward system.
- **Quick Delivery:** Enables early access to core features while continuing development.

For "Quiz & Earn," where features need regular adjustments and scalability is key, Agile is the most suitable SDLC model to ensure successful and timely delivery.

Table 1: Comparison of SDLC Models for Quiz & Earn: A Dual-Mode Quiz App

Priority	Criteria	Waterfall	V-Shape	Iterative	Spiral	Agile	Prototype
5	Well known requirement	No	No	Yes	Yes	Yes	Yes
3	Technological knowledge	Yes	Yes	No	No	No	No
5	Efficiency	Yes	Yes	Yes	No	Yes	No
6	Risk analysis	No	Yes	No	Yes	Yes	No
3	User testing ability	No	No	Yes	Yes	Yes	Yes
5	Dependability and Security	Yes	Yes	No	No	Yes	No
3	Time consuming	Yes	Yes	Yes	Yes	No	No
Total 30	Over all	16	22	16	17	24	08

References

[1] Barry W. Boehm. A spiral model of software development and enhancement. In *ACM SIGSOFT Software Engineering Notes*, volume 11, pages 14–24. ACM, 1988.