

PERSONALIZED HEALTH AND WELLNESS ASSISTANT



SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC) MODEL

Course Title – Software Engineering

Slot – G2 + TG2

Name – Vansh Kumar Payala

Registration No. – 22BAI1386

Introduction –

The ‘Personalized Health and Wellness Assistant’ will serve as a new generation mobile application that will include artificial intelligence and machine learning components to suggest a recommended health and wellness plan based on data inputted by the user. This is a new concept which will include a chatbot and will assist the users with medication and prescription guidance; therefore, the usage of the app will be enhanced while appropriate health support will be provided. The most important one is to educate and give specific recommendations concerning the state of one’s health and help to make the right decision.

To accomplish the mentioned above proposed objective of a well-structured and efficient manner of development, Software Development Life Cycle, namely the Incremental Model has been chosen. This way the application is develop in phases including this feedback of users and can be improved gradually by incorporating more features. Thus, when realizing the project in stages, it is also possible to ensure that each of the elements is functional and the final outcome is a usable application that will meet the needs of end-users. It also shows the concrete description of Incremental Model in the framework of the ‘Personalized Health and Wellness Assistant’, as well as naming all the activities of each phase and their outcomes.

Incremental Model –

The Incremental Model is a model of software development that breaks down a project into smaller parts that can be measured. Furthermore, each increment is a different type, which increases the previous one but provides new options and opportunities. This makes it possible for developers to do adequate concentration in giving out a working product at the end of every cycle with some essential features being developed in the early cycles. Here we have given the decomposition of the project into subarrays to tackle complexities and risks and to optimize changes based on the usage of the project which can be easily taken up by the development team.

This model is very helpful especially when the projects are cyclic and the users are involved in providing feedback. In each phase, developers may also have the feedback of the users as well as the stakeholders and then the program developed as well as improved before moving to the next level. This results into an iteration process that enhances the final design to be in line with the users’ expectations. The Incremental Model is also very flexible and it is easy to make adjustments in one of more phases of the development. It is most appropriate to use in the systems where their requirements may change in the future, which makes this strategy customer-oriented.

Phases of the Incremental Model –

The development process using the Incremental Model is divided into several phases, each contributing to the progressive enhancement of the system. These phases include:

- Requirement Analysis
- System Design
- Implementation
- Testing
- Deployment
- Maintenance

Phase 1: Requirement Analysis

As a part of Requirement Analysis phase all the requirements related to ‘Personalized Health and Wellness Assistant’ are acquired and analysed. Once you comprehend the users’ needs, this phase outlines the system characteristics to enhance general usability of the application.

Tasks:

1. User Research:

- Schedule one on one meetings and questionnaires with the intended target market keeping health and fitness in mind.
- Figure out major drawbacks and suitable options to know what clients want concerning nociceptive modulation.
- User personas need to be created to portray various segments of the user population.

2. Market and Solution Analysis:

- Explore the existing applications in the health and wellness context to have an understanding of the best performing as well as the shortcomings.
- Investigate the market to look for gaps where the firm could create a niche for itself.
- To determine competitive features and advantages, benchmarking should be carried out against the better solutions.

3. Feature Definition and Prioritization:

- Discuss basic components that include healthcare advice and prescription based on user's data, simple communication with virtual assistant, and handling user's information.
- Sort features according to the findings derived from the research conducted on users and the market to meet project objectives.
- Make sure that the aspect addresses the users requirement and brings value addition to the application.

4. Requirements Documentation:

- Client requirements should be carefully collected and put together in the Requirements Specification Document that would emphasize on both the functional and non-functional requirements.
- Include User Stories and Use Cases to describe how users will engage with the application in the future, to inform the further work.

Outputs:

- Requirements Specification Document: In this case a document containing all aspects of the system requirements.
- User Stories and Use Cases: Specifications of the users 'use and engagements as well as usage situations to inform the development process.

Phase 2: System Design

During System Design, we make detailed descriptions of the applications layout, storage strategy, and the retrieval system, and the graphical user interface design. It brings about clear definition of the structure development that can help in developing the next phases of the project.

Tasks:

1. Architecture Design:

- Identify and discuss the extent of the system and its parts also the manner in which they are integrated.
- Passwords, accounts, roles, products etc. are some of the issues that should be addressed in client-server model, API locations and logic of the system.

- They can use it to define the overall structure of the systems as well as some of the specific features of the application.

2. Database Schema Design:

- Establish frameworks for other structures that shall provide the necessary canvass and hub of the data that will be collected and stored.
- Avoid mistakes in how the schema is set up: When the application is in production, make sure that it is as performant and scalable as possible, and that it is properly structured.
- Emphasis on how creation of tables and their relationships, and indexes are implemented.

3. Integration Planning:

- Define integration points for the chatbot, detailing how it will interact with the AI model and user interface.
- Develop integration specifications to ensure smooth interaction between system components.

Outputs:

- System Architecture Diagram: A representation of the approximate steps and numbers to be used in the course of app development.
- Database Schema: Documentation plan that covers such areas as listing of all the tables in the database together with its relations and indices.

Phase 3: Implementation

Implementation is the development of the actual application for the new system as it is being described in the model. This phase is actually split into many micromanagers to each of which corresponds a number of distinct features and functions.

Tasks:

1. Development Environment Setup:

- Install the required tools and libraries that are needed for the developmental process.

- Ensure cloning of the development environment in order to make coding to be easy.

2. Core Feature Development:

- Develop and integrate such mandatory components like the user registration or the ability to log in into the application providing secure access to the further sections of the application and its features.
- Extend the model with product recommendations for providing health and wellness recommendations with reference to the data of users.

3. Chatbot Integration:

- Connect the chatbot to offer medication and prescription advice according to the diagnoses given by the AI model.
- Coordinate well the conversation between the chatbot and the artificial intelligence model.

4. Mobile Application Development:

- Develop the mobile application with the help of Kotlin.
- Implement the app to ensure that the ends user interacting with the application is on an Android based device.

Outputs:

- Source Code: Application's source code, main functions and capabilities / Full code of the application, the main functionality of the program.
- AI Model: The developed prototype system that can recommend the health and wellness plan based on the user's context.
- Integrated Chatbot: It focuses the chatbot with the AI model on medication and prescription consultation.
- Mobile Application: A working mobile application for Android Implemented in Kotlin.

Phase 4: Testing

Verification is solely confirmation where each subsequent step is anticipated to produce substantive results and it meets the user's expectations. This phase is crucial because of

problem solving and identification and is facilitated by the formal testing carried out in this phase which is done at several levels.

Tasks:

1. Test Planning:

- Make sure that the application is extensively, comprehensively and solely conforming to the required and expected quality by overdramatizing the formulation of good test cases as well as good test plans.
- Develop a document of how testing is going to be done and scenarios that need to be created to test various characteristics of the program.

2. Unit Testing:

- Moreover, do some form of integration test in order to assess the capacities of the components after they have been integrated with other components.
- In this stage, the search for defects and repairing of the defect is carried out before the compound is mixed with another compound(s).

3. Integration Testing:

- Making sure that all the parts are fully connected in integration testing to fully be convinced.
- Ensure the related data is transferred between the divided parts of the application as should be.

4. User Acceptance Testing (UAT):

- Recruit real users to the project to be used in testing the application in terms of usability in addition to functionality.
- Determine the heuristic usability problems and gather more information for change of improvement.

Outputs:

- Test Cases and Plans: A clear indication of the specifics of the testing schedule and noting a clear list of the type of tests that is expected to be conducted.
- Test Reports: Aspects of the test(s) that were conducted, the issues observed, and the made decisions.
- User Feedback: However, for authentic customer identification some sections may be appropriate and necessary to fine-tune and even boost.

Phase 5: Deployment

The Deployment phase entails deploying to the production the increments that were developed in the previous phase. Each increment is a release of software and a release is posted when the software is considered to be stable enough and performing up to expectations.

Tasks:

1. Production Environment Preparation:

- Get all the structures in a state of readiness for scripting such as the servers and/or the databases.
- Perform an environmental check to see if the chosen deployment strategy is compatible with the environment and meets performance standards.

2. Incremental Deployment:

- Place each increment one at a time in the operation environment.
- Must be keen on the deployment process in order to retain stability as well as performance.

3. Performance Monitoring and Feedback Collection:

- Ensure that the applications function as required and are reliable once released to the public.
- Engage the users so you can easily notice any issue or improvement that can be made.

Outputs:

- Deployed Increments: Organic components that already, somehow, got exposed to the productivity environment.
- Deployment Documentation: This involves procedures and settings that are required for the deployment to be done and also serving of the step by step procedures.

Phase 6: Maintenance

The Maintenance phase takes measures to make sure that the developed application completes the required function, is updated on time, and is relevant to the users. This consists of addressing previously identified problems, improvements, and new versions with regard to the users.

Tasks:

1. System Monitoring and Issue Resolution:

- Closely monitor the system performance and review various logs to be able to handle conditions that make their way into the system once in a while.
- Occasionally it will involve firmware release and bug fixes and patches in case there is a need to fix the application and improve on the stability and security of the application.

2. Enhancements and New Features:

- Nurture ideas for improvement and new components that are created by the users or due to the new requirements emerging during system development.
- Ensure that the changes are consequential in meeting or improving the user need and isn't simply cosmetic but improves the application in some capacity.

3. User Support and Training:

- From here, we need to engage the users and constantly assist them, or even train them, on how to effectively use the app.
- Make and promote some of the papers as well as toolkits that can be helpful for users.

Outputs:

- Bug Fixes and Patches: Repairs carried out on areas that have been identified to have issues within the compartment used in production.
- System Enhancements: This feature covers the application in as much as it describes new features and changes in terms of improvement.
- User Support Documentation: Guides and resources to assist users in utilizing the application effectively.

Conclusion –

Utilizing the Incremental Model for the development of the 'Personalized Health and Wellness Assistant' allows for growth and a systematically organized manner of designing an application from the perspective of the user. This model encourages incremental work, and thus feedback is made and incorporated at every stage of the development. Every increment adds useful functionalities and improvements into the application and guarantees that it will develop as the requirements of the users and tendencies of the markets differ.

By adopting this model, the process of adopting users feedbacks, improving the functions, and aligning the entire application with the users' expectation becomes systematic. It also allows for the infinite loop with the objective of a final work of high quality and efficiency in the product to present application that still has the capacity to offer constant improvements and meet emergent needs of the health and wellness sector.

To sum up, it is possible to conclude that within the context of the strictly defined Incremental Model, it is possible to achieve the goals of designing a highly effective health and wellness solution that would take into account the needs of all user types, as each phase of the process contributes valuable input to final result.