“SuppleMate” App
Software Project Management Plan (SPMP)

Team:

- Gregy Thomas Kokkaparampil – Project Manager
- Shuo Li– Assistant Project Manager/ Software Engineer
- Siddharth Santhanakrishnan – Software Engineer
- Wone Eui (Ella) Jung – Software Engineer

Table of Contents

1. Overview **4**
1.1 Project summary 4
1.1.1 Purpose, scope, and objectives 4
1.1.2 Assumptions and constraints 4
1.1.3 Project deliverables 5
1.1.4 Schedule and budget summary 5
1.2 Evolution of the plan 5
2. References **5**
3. Definitions **6**
4. Project organization **6**
4.1 External interfaces 6
4.2 Internal structure 9
4.3 Roles and responsibilities 9
5. Managerial process plans **11**
5.1 Start-up plan 11
5.1.1 Estimation plan 11
5.1.2 Staffing plan 12
5.1.3 Resource Acquisition Plan 12
5.1.4 Project staff training plan 12
5.2 Work plan 12
5.2.1 Work activities 12
5.2.2 Schedule allocation 13
5.2.3 Resource allocation 13
5.2.4 Budget allocation 13
5.3 Control plan 13
5.3.1 Requirements control plan 13
5.3.2 Schedule control plan 13
5.3.3 Budget Control Plan 14
5.3.4 Quality Control Plan 14
5.3.5 Reporting plan 14
5.3.6 Metrics collection plan 14
5.4 Risk management plan 14
5.5. Closeout plan 15
6.2 Methods, tools, and techniques 16
6.3 Infrastructure plan 16
6.4 Product Acceptance Plan 16
7. Supporting process plans **17**
7.1 Configuration management plan 17
7.2 Verification and validation plan 17
7.3 Documentation plan 17
7.4 Quality assurance plan 18
7.5 Reviews and Audits 18

7.6 Problem resolution plan 18
7.7 Subcontractor Management Plan 18
7.8 Process improvement plan 18

1. Overview

1.1 Project summary

The Software Project Management Plan lays out the details of the management plan that will be followed to develop the SuppleMate project. This plan outlines the project's development cycle, organization, process plans, and projected timeline.

1.1.1 Purpose, scope, and objectives

The Software Project Management Plan (SPMP) will relay all the details regarding the development plan and the development cycle. It will assign each member roles and duties regarding the development, and how and what methods they will use to finish their tasks. It will also assign deadlines so that the project will develop on time. It will follow the specifications declared in the most current version of the SRS that was agreed upon by all team members and the client. The MVP prototype will contain the UI design. It will be able to navigate through different pages.

The SuppleMate is designed to help users manage their dietary supplements and vitamins effectively. It allows users to input and track their supplement intake, compute the exact dosage required each time, set dosage reminders, and receive personalized nutritional/ supplement recommendations based on their health goals and nutritional needs. The app provides insights into the benefits of each supplement, tracks expiration dates, and offers a comprehensive overview of the user's overall supplement regimen.

1.1.2 Assumptions and constraints

The list of all the assumptions and constraints

- Team members will attend all meetings
- Team members will meet all the deadlines
- Team members will follow the requirements specified in SRS
- The application is being designed to run on the web and fit different screen size
- Team members will work on the project outside the class to finish it on time

1.1.3 Project deliverables

- Working Executable web application
- SRS, SPMP, Documented Source Code, Software Architecture, Maintenance Manual

1.1.4 Schedule and budget summary

There is no budget for this project. According to the course outline, the working prototype will be delivered by week 10.

1.2 Evolution of the plan

The team met with the client on October 28th, 2023, and discussed the expectations and functionality of the application. Therefore we updated the requirements of the prototype and made sure the client was up to date and knew the process within the project.

2. References

[1] IEEE Software Engineering Standard Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.

[2] IEEE Std 1058-1998 for Software Project Management Plan

3. Definitions

Programming Language: A vocabulary and set of grammatical rules for instructing a computer or computing device to perform specific tasks.

SRS: This document, the software requirements specification (SRS) document lays out the functional and non-functional requirements of the SuppleMate App.

UML: Unified Modeling Language (UML) is a standardized modeling language enabling developers to specify, visualize, construct, and document artifacts of a software system.

Use case diagram: A representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.

User: Someone who uses the application or software specified in this software requirements specification.

API: Application Programming Interface – refers to a set of routines, protocols, and tools that are used in the building of software applications.

MVP: A Minimum Viable Product (MVP) is the most basic version of a product that contains only essential features to meet the needs of early users. It is designed to gather feedback, validate ideas, and quickly bring a functional product to the market.

4. Project organization

4.1 External interfaces

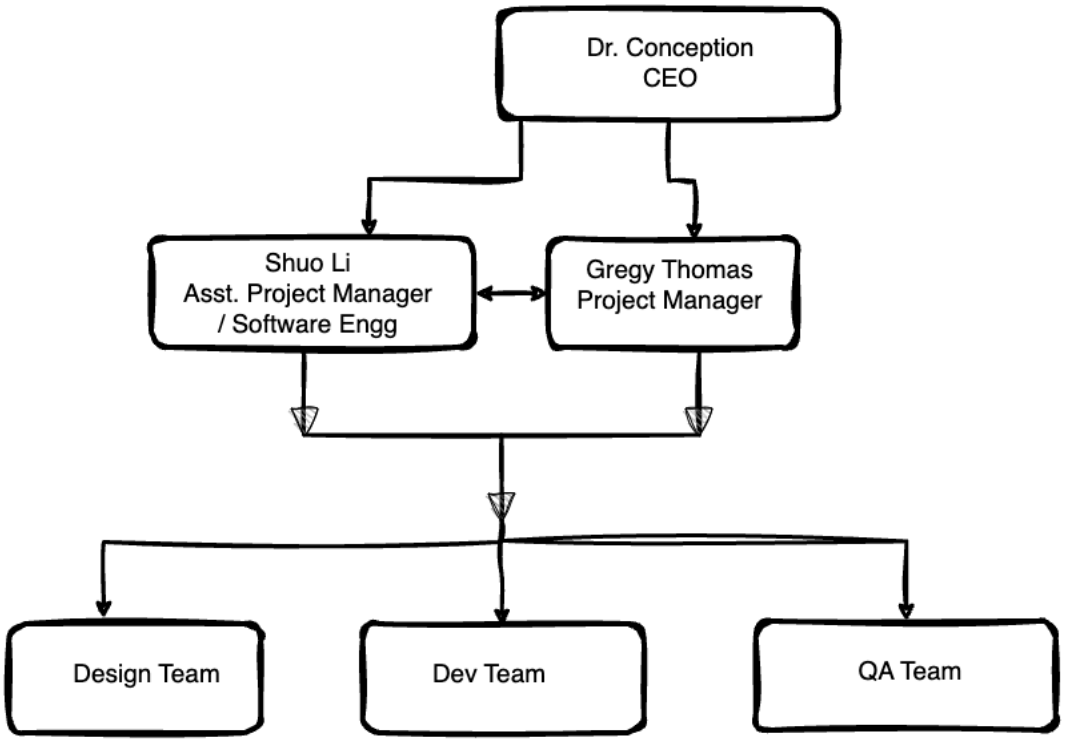

Figure 1 Project organization

The CEO: The CEO provides the overall vision and strategic direction for the SuppleMate project. Also responsible for securing funding and resources. Dr. Conception sets the high-level goals, approves budgets, and is periodically updated on project progress by the project manager and the development teams.

Project Manager: Oversees the day-to-day management of the SuppleMate project. They define product requirements, manage the development team, and ensure project milestones are met. Collaborates closely with the CEO to align the project with the company's goals. Communicates the project's progress and needs to the CEO and coordinates with the rest of the teams.

The Associate Project Manager: Assist in project management tasks and UI development. Works closely with the Project Manager, to manage project tasks and timelines. They also interact with the design and development teams to ensure smooth UI integration.

Design Team: Responsible for creating the user interface (UI) and user experience (UX) of the SuppleMate app. They design the app's look and feel, ensuring it is visually appealing and user-friendly. The Design Team collaborates with the Project Managers to understand design requirements and provide design assets to the development team for implementation.

Dev Team: Responsible for turning the design and product requirements into functional software. They build and maintain the SuppleMate app, ensuring it works as intended. The Dev Team closely collaborates with the Project Managers to understand development priorities, milestones, and technical requirements. They also work with the Design Team to implement the UI.

QA Team: Responsible for testing the SuppleMate app to identify and report any issues, bugs, or usability problems. They ensure the app functions correctly and meets quality standards. The QA Team works closely with the Development Team to report and track issues, and they collaborate with the Project Managers to ensure that bug fixes and improvements are prioritized and addressed.

4.2 Internal structure

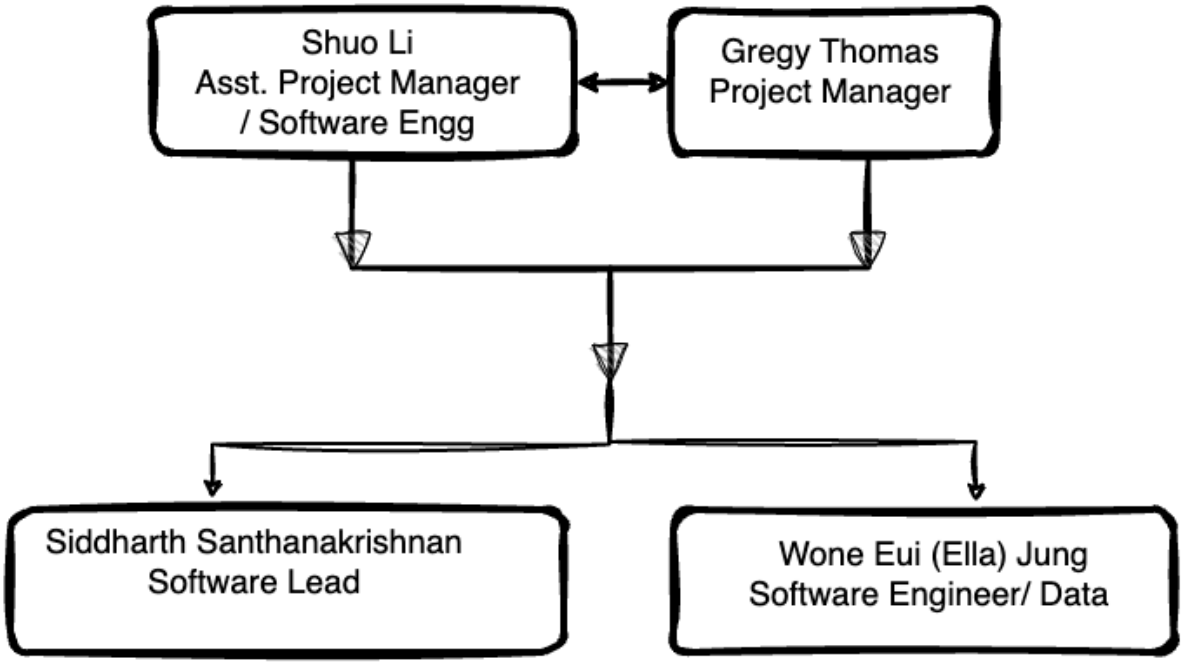

Figure 2 Team Structure

The project manager and assistant project manager work together to facilitate progress in completing the prototypes.

4.3 Roles and responsibilities

1. **Gregy Thomas Kokkaparampil**: Overseeing the entire product development process. Responsible for defining the product vision, setting strategic goals, and guiding the development team to bring SuppleMate to life. Responsibilities include:
    - gathering and prioritizing user requirements.
    - creating product roadmaps.
    - managing the project timeline and resources, and
    - ensuring that the final product aligns with the intended vision.

    - Act as the bridge between the development team, stakeholders, and end-users, making critical decisions to meet business objectives and deliver a successful, user-friendly supplement tracking app.

2. **Shuo Li**: Monitors team progress in accomplishing development goals and manages documentation resources. In the event of the project manager's absence, APM can take over duties and responsibilities associated with the Project Manager. Also has direct involvement in the UI development of the application.
    - Gather and specific user requirements.
    - Communicate with clients and development team
    - Draft specification documents.
    - Track and keep the process of all team members, make sure the project process meets timeline and requirements.
    - Design and develop User Interface through Figma.

3. **Siddharth Santhanakrishnan:** Heads the development team responsible for building the SuppleMate app. Oversees and leads the development efforts, ensuring that the app is built according to design and functionality specifications. They coordinate with the development team to ensure the app is developed successfully and on time.
    - Ensure the app is built in accordance with design and functionality specifications.
    - Coordinate with the development team to ensure timely and successful app development.
    - Contribute to technical decision-making and problem-solving.
    - Maintain code quality and development best practices.

    - Assist in resolving technical challenges and issues as they arise.
    - Collaborate with the Project Manager to meet project milestones and objectives.

4. **Wone Eui Jung**: In charge of building databases, importing data, accumulating historical data, and validating data correctness.Transfer data in base granular level for supporting reports. Create stored procedure of the database and export the data with python dataframe
    - Collect supplement intake guide line and supplement product containing data.
    - Create a database with MongoDB after cleaning up collected data.
    - Create Python code that delivers data into a data frame.
    - Create NoSQL procedure that can filter and produce results upon request based on logic.

5. Managerial process plans

5.1 Start-up plan

5.1.1 Estimation plan

- Discuss with the client all the specifications for the application
- Research technologies required to meet all the requirements
- Create guidelines for working prototype in a given time
- Estimate the deadline for each task to make sure the prototype can work on time

5.1.2 Staffing plan

The members willingly organized the team together after understanding every member's background and discussing the advantages and disadvantages of each member of the team.

5.1.3 Resource Acquisition Plan

All the software tools involved in the development of the application are free.

5.1.4 Project staff training plan

All the team members will complete the course materials, and each member of the team will learn additional technologies that they might need to finish the project.

5.2 Work plan

5.2.1 Work activities

Prototype 1: This MVP prototype will build an application that has a functional user interface that allows users to build a new profile, or log in to their account. A functional homepage where users can see their nutrition details. A functional reminder which can remind users to take their supplement on time with the right dose.
Prototype 2: The second prototype will provide a recommendation function where users can get personalized supplement recommendations based on individual needs.
Documentation: All members will make sure that all the code and document that they write is properly documented.

5.2.2 Schedule allocation

10/30/2023	11/05/2023	Prototype #1 Development
11/06/2023	11/12/2023	Prototype #1 Development
11/13/2023	11/19/2023	Prototype #1 Development and testing
11/20/2023	11/26/2023	Prototype #1 Development and testing
11/27/2023	12/02/2023	Prototype #1 Final testing and delivery
12/02/2023	12/10/2023	Prototype #1 Final testing and delivery

5.2.3 Resource allocation

Each member has to research the resources they need to use.

5.2.4 Budget allocation

No budget has been allocated for this project.

5.3 Control plan

5.3.1 Requirements control plan

Each team member is required to attend the meetings related to the project. Additionally, each member is required to document their code, follow the guidelines decided in the SRS, and meet each deadline as well. Any unexpected issues, technical difficulties, or requests by the clients will be assessed by the Managing team and decided upon.

5.3.2 Schedule control plan

Aside from the regular class meeting time, the team might be required to meet outside the class as well to finish and deliver the product on time. Managers will keep a close eye on the

process of the project and make sure that each member is completing their tasks on time.

5.3.3 Budget Control Plan

There is no budget allocated to the CoyoteQuest team. However, we will make sure that the project is efficient enough so that it doesn't overtax the budget.

5.3.4 Quality Control Plan

All team members will constantly perform a quality check on the software at least once a week to make sure that the project meets all the expectations. Also the client will be present during the development of the application and therefore will be informed of the quality of the application.

5.3.5 Reporting plan

The project manager will update the processes of the project to the client.

5.3.6 Metrics collection plan

All team members will make sure that each week every developer completes their tasks so that the project is on track and progressing as expected. Each member will make sure that their code is efficient and meets all the standards.

5.4 Risk management plan

Development

- The team will meet regularly to make sure that the production is not stopped at one particular point.
- There will be scheduled deadlines that everyone will be following to make sure the project is not delayed

● Each team member will be kept up to date, and inform other team members of any changes or difficulties that might affect their ability to complete the task they are given on time.

Project Failure

● If the technology is not suitable for the project, the team will discuss it with the client and make sure there are other ways to meet the client's requirements.

● Every member is expected to help carry the weight of any struggling members.

5.5. Closeout plan

The team will submit the deliverables to Github. The team will also present the application at the end of the course.

6. Technical process plans

6.1 Process model

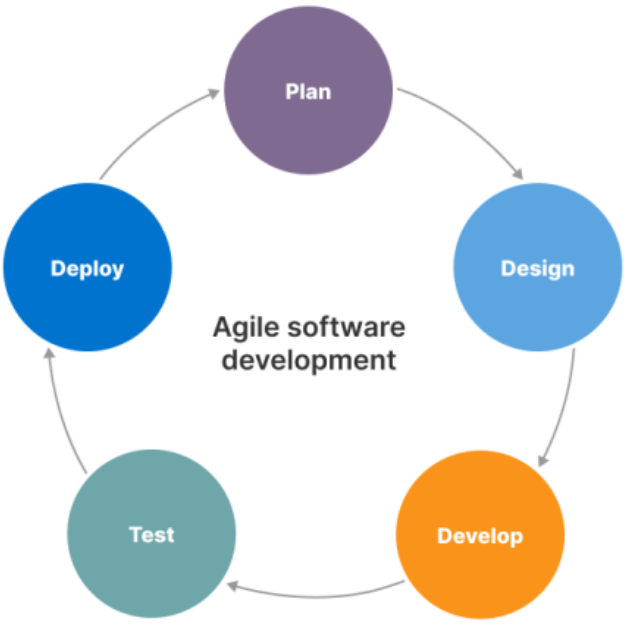

Figure 3 Process Model

Agile software development refers to a collection of methodologies and frameworks that promote adaptive planning, evolutionary development, and continuous delivery. Instead of focusing on a final output — the finished product — agile teams prioritize working in short increments with frequent release cycles to deliver value faster. The procedures will be repeated several times if needed. All the procedures must be done within a six-week span.

6.2 Methods, tools, and techniques

Method: Incremental Development Model
Tools: IntelliJ, Java, Springboot, Visual Studio, Anaconda Python, MySQL, Figma for UI
Techniques: Sprint ceremonies on a weekly cadence

6.3 Infrastructure plan

There is no need for an infrastructure plan at the moment. The server will be either hosted via Amazon services or a local machine.

6.4 Product Acceptance Plan

The client will be kept throughout the development process. All members in the team will keep testing the project and make sure the prototype will work and be acceptable.

7. Supporting process plans

7.1 Configuration management plan

We are using GitHub for the configuration. It will contain all the changes made to the code and will allow for a smoother configuration.

7.2 Verification and validation plan

Verification and validation involve periodic testing of the app, including unit, integration, and system testing. Bugs and errors discovered during testing are documented and reported to managers.

- **Unit testing** requires each programmer to test their assigned software component for code quality and fluidity.
- **Integration testing** involves testing all software components together to identify errors.
- **System testing** evaluates the prototype software with project managers, the client, and the quality assurance team, ensuring a quality product before final delivery.

Any required changes or errors found trigger a repeat of the Scrum cycle, starting with client planning and proceeding through unit, integration, and system testing.

7.3 Documentation plan

The managing team will prepare the SRS and SPMP. The developers will write the documentation of design and architecture.

7.4 Quality assurance plan

Each team member will make sure of the quality of the parts they work on.

7.5 Reviews and Audits

During the development and testing phases, every member of the team will test and report any deficiencies in the app. Design flaws or bugs will be reported and documented for immediate fix or future review.

7.6 Problem resolution plan

Each member of the team will keep other team members up to date on any issues that they might encounter. Afterwards, the team will discuss how to handle those issues and make sure that the project is completed. The team will also make any changes necessary to make sure that the project runs as smoothly and as efficiently as possible.

7.7 Subcontractor Management Plan

We have no subcontractors.

7.8 Process improvement plan

To improve the process and development of the software, it is crucial to write good documentation of the source code that will be provided. Some features cannot be implemented in the given time at the moment and therefore have been pushed back to the next stage of development. Writing quality code

now, documenting it, and creating a good maintenance manual will help with future development. To help improve code production or code quality, every member must report to the managers on their progress.