Project Management Plan

Group 5 - Simulation Training Program for Extracorporeal Membrane Oxygenation
FIT4002 - Software Engineering Industry Experience Studio

Version 3.0
Document as of 3 Aug

Team Members

Soo Ying (Celine)	Product Owner	celine.yeap@monash.edu
Denzel	System Architect	denzel.long@monash.edu
Jason	Scrum Master	jason.toh@monash.edu
Jeremy	Tech Lead	jeremy.leckning@monash.edu
Mi Jia (Micah)	Quality Assurance Engineer	MiJia.Looi@monash.edu
Philip	DevOps Engineer	philip.chen@monash.edu

Contents

1.0 Summary

ECMO, cardiovascular modelling

2.0 Project Overview

Project Title: Simulation Training Program for Extracorporeal Membrane Oxygenation

The Simulation Training Program for Extracorporeal Membrane Oxygenation (ECMOSTP) is an educational web application to provide training on the operation of the life support system, ECMO, thereby increasing the chances of survival of patients in critical conditions.

The education suite provides a library of scenarios available for training, using 3D scenario simulations and a numerical model of the ECMO machine and cardiovascular system. The current simulation is implemented in the MATLAB/Simulink environment, and will be migrated to the new system built in Unity3D.

In the future, the project is envisioned to extend and support e-assessments and e-invigilation functionalities.

3.0 Vision

Vision statement
The vision of the *Simulation Training Program for Extracorporeal Membrane Oxygenation (ECMOSTP)* is to provide an accurate and realistic way that facilitates ongoing learning and skills retention to ECMO, with the mission to improve ECMO patient outcomes through the web-app training simulator.

This system achieves this by simulating different patient scenarios from an interactive scenario library, with the ability to interface with a virtual ECMO machine. The success of this system will greatly promote the exposure of ECMO to the clinicians, which is vital in helping an ECMO patient. This project can be summarised as:

> **For** clinicians trainees and trainers
> **Who** needs the experience and training on the operation of ECMO machine as well as facilitating medical education
> **The** Simulation Training Program for Extracorporeal Membrane Oxygenation
> **Is an** interactive textbook
> **That** provides a series of simulated ECMO patient scenarios and the ability to interface with a virtual ECMO machine with clear and directed learning outcomes
> **Our** product, which is a web-based application, will be able to help ECMO staff acquire skills needed to operate an ECMO machine even with the lack of bedside ECMO exposure. Our product also facilitates a realistic way to train both junior and senior ECMO staff that allows ongoing learning, competency maintenance and skills retention, resulting in optimal patient care and improved outcomes.

4.0 Deliverables

The minimum viable product:

- Source codes (MVP) that is ready to be deployed in 2D:
    - Scenario simulation (proof of concept with example)
    - Framework to modify scenarios library
    - Textbook functionality with layouts and is editable
- Numerical model (ECMO and cardiovascular model) from Pulse Engine to be hosted and communicable via API
- Documentations for deployment, maintenance of software, user manual

Stretch goals
- Integration of e-Assessment and e-Vigilation

Further extensions and applications of product:
- Training progress reports
- 3D models in Unity3D

4.1 Product Backlog

The product backlog consists of features required in the product, and deliverables expected by the clients. Items listed in the backlog can include features to be developed, bugs, non-functional requirements, documentations, and is ordered by its priority. The items are listed in the format of:

As a (role), I want to (feature), so that (purpose)

Each item is labelled with:
- Priority: On a scale of Lowest, Low, Medium, High, Highest priorities.
- Estimation: On a scale of 1-5, with 5 requiring the most effort in completing the task
- Status: To do, In progress (currently working on), Done (item is completed)
- Dependencies (if any): ID of dependent item

This list will continually grow and change as more requirements are gathered during the development. This backlog will be reviewed at the end of every sprint to add new requirements, modify, redefine, delete existing requirements.

4.2 Sprint Backlog

Sprint backlog consists of items to be completed by the team in each sprint. During a sprint meeting, the team must be sure of steps required for tasks to be completed before having items moved into the sprint backlog. This is planned according to the team's capacity and the estimation of the backlog items. Sprint backlog items are assigned to members responsible for completing the tasks.

The sprint backlog is updated as new information of tasks is made available. A sprint burndown chart can be generated with this backlog.

4.3 Roadmap and Timeline

A roadmap is used to describe how the product will be developed over time. This roadmap is shared across teams in eSolutions. Releases of new features and functionality are detailed in the roadmap.

A timeline describing when the team aims to deliver each feature is used and maintained on a spreadsheet. Unit assessments are also included in the timeline so that the team is able to estimate its capacity while balancing commitments outside of the project.

Timeline

5.0 Project Context

5.1 Stakeholders

There is a need to balance between satisfying stakeholders of this project and its feasibility, as it is impossible to always keep all stakeholders satisfied. This can be done by carefully managing and prioritising the project stakeholders. Stakeholders of this project are managed in groups according to their influence and interest towards the project.

Stakeholder mapping:

<table>
<tr><td rowspan="3">Influence</td><td>High</td><td>Keep Satisfied:
Product end-users
Clinicians - Raymond Li (Med student, involved in training simulation)</td><td>Manage Closely:
Clients
CREATElab: Andrew Stephens
Monash University eSolutions: Paul, Irwyn
Our team
Jason (Scrum Master)
Philip (DevOps Engineer)
Soo Ying Yeap (Product Owner)
Mi Jia Looi (QA)
Denzel (System Architect)
Jeremy (Tech Lead)</td></tr>
<tr><td>Low</td><td>Monitor:
FIT4002 teaching staff:
Waqar Hussain (Supervisor)
Yuan-Fang Li (Unit coordinator)
In the list of project supervisors: (no direct contacts made)
Dr. Michael Seman
Dr. Shaun Gregory</td><td>Keep Informed
eSolutions staff: Derek, Samuel</td></tr>
<tr><td></td><td>Low</td><td>High</td></tr>
<tr><td colspan="4">Interest</td></tr>
</table>

Manage closely: this group of stakeholders have direct influence on the project in terms of decision making for the project. Regular engagement and communication is required to understand the expected outcomes of the project. Should be involved in project planning meetings (sprint planning & review, release planning, establishing definition of done).

Keep informed: Low influence but highly interested in the product. These stakeholders should be regularly informed of the progress and their opinions are valued, however not at all times considered in the event of conflicting interests.

Keep satisfied: Require updates of progress and opinions are valued for certain key decisions (e.g. opinions of end users in user experience of the product)

Monitor: Keep them in the loop by informing them of important news and decisions.

5.2 Process Model

5.2.1 SCRUM and our use of SCRUM

- Reasons for choosing SCRUM

    - Team is familiar with Scrum
    - Scrum is agile in nature, important for a project with a wide scope like this one. We can adapt and focus on various aspects of the system for the next sprint to accommodate client's requirements
        - Scrum is built to handle change, agility with Scrum allows us to iterate quickly and fail quickly
        - Result in mistakes that are less likely to severely impact product delivery
        - Allows the entire team to learn from each other's experiences and mistakes
    - Scrum allows for high levels of teamwork and collaboration, a important learning outcome for the FIT4002 unit
        - The Scrum Master is responsible for the upkeep of morale within the team
        - As well as facilitating group problem solving and ideation.
    - Sprint Retrospectives and subsequent Sprint Planning for the next sprint allows for an organic continuous improvement through team reflection
- What aspects of SCRUM do we follow, and what did we not follow/change?
    - We have kept the essence of Scrum and it's agile nature
    - However, due to the project being a student led project, we have decided against daily stand-up meetings
    - We have also configured the sprint length to be two-week long sprints

5.2.2 Tools and techniques

The application design architecture of choice for this project is following the Microservices approach, where said architecture allows us to integrate various services of different purposes in this case, we need to feature a heavily computational service, a 3D model service and various interactive services as well. Therefore, being able to split such independent services which originate from different codebases seamlessly into the main application, would promote a more flexible framework which is desired by the client for future extension of functionalities.

The design approach for the front end of the web application would be following the Model View Controller design pattern with React and Material UI as our framework. In terms of backend, Express by Node.js would be the tool of choice to handle backend calls to the database as well as API calls for calculations. For our 3D model of the ECMO machine, we would use Unity to handle the load.

This is further detailed in the System Architecture Vision document.

5.2.3 Task and resource management

tracking progress and time - burn down chart, Jira etc.
(Can say mentioned earlier in deliverables section)

5.2.4 Roles and responsibilities

- Member roles and what are their responsibilities

The team has defined six internal roles. This includes, a Product Owner, a System Architect, a Scrum Master, a Technical Lead, a Quality Assurance Lead, and a DevOps Lead. Each of these roles includes its own defined duties and responsibilities, which will be addressed in the table below. On top of these internal roles, all six members of the team are also concurrently Software Developers. As software developers, members of the team could be assigned technical tasks which could include the implementation of backend systems, frontend development, middleware components, database management and hosting services.

Role	Member	Responsibility
Product owner	Soo Ying Yeap (Celine)	● Stakeholder management: ○ Communicates stakeholder requirements to the team ○ Balance needs of stakeholders ● Release management ○ Planning when things are delivered ● Manage requirements documents
System Architect (model lead :))	Denzel Long	● Planning and designing system architecture ● Make decisions regarding choice of technologies together with Technical Lead ● Updating SA documents accordingly
Technical Lead	Jeremy Leckning	● Planning and assessing suitability of tools used in the project ● Make decisions regarding choice of technologies together with System Architect ● Works with SA
Quality Assurance Lead	Mi Jia Looi (Micah)	● Identify risks of the project and planning for risks through mitigation strategies ● Upkeep of Quality Assurance Plan using a risk register ● Ensuring quality assurance plan is followed through
DevOps Lead	Philip Chen	● Planning of CI/CD workflow ● Release delivery gatekeeper ● Ensuring workflow is followed through by the team
Scrum Master	Jason Toh	● Ensure adherence to process models and the Scrum-ness of the project ● Ensure fairness of workload and upkeep morale of the team ● Backlog management

5.2.5 Client Communications

There are four main channels of communication between our client and the development team.

1. Weekly Client Meetings
2. Slack Channel
3. Product Backlog/Jira Board
4. Direct Email

Weekly client meetings are headed by our Product Owner, delivering questions about requirements gathering, technical inquiries and general team concerns. It is usually held as a forum form whereby any member of the team is encouraged to speak out about their ideas and seek clarification from the client about their preferences. Live demonstrations of the product are to be delivered during the Weekly Client Meetings at a bi-weekly schedule. This allows the client to provide immediate feedback for the team to make adjustments before the next demo. This is a feature of scrum which allows teams to deliver higher quality products within a

shorter amount of time. Our Weekly client meetings are also recorded, which allows us to revisit parts of the meeting to gain more clarity about the project after the meeting.

Next channel of communication is our Slack channel with the client. Our slack channel provides a means of quick and easy informal communication with the client. A developer working on the front-end user interface, of the back-end architecture, could quickly send a sketch of a prototype design and get immediate feedback from the client. The slack channel is the quickest way to reach our client in scenarios where time is critical.

Also, we have our Jira board, containing our product backlog. Here we detail our user stories, and also the task owner for a particular user story. This allows our client to see a birds eye view of progress, and how detailed and intricate a seemingly simple request may be.

Lastly, we can communicate with our client through email. This is our most formal form of communication with the client and is typically used in cases where we require an official response such as topics like project scope and software requirements.

5.2.5 Methods of contact

Due to the restrictions caused by COVID-19, some of our team members are not currently in Melbourne. Hence, in addition to face-to-face meetings/presentations when necessary, document-sharing and online collaboration will involve the use of the following tools for the project:

Collaboration Tool	General Purpose
Discord	Team-level text messaging and voice chat for team meetings
GitLab	Git-repository manager for artifacts
Zoom	Video meeting with clients and supervisor
Google Drive	Team document management and meeting recordings
Slack	Text messaging with team members and clients
Jira	Agile project management tool (e-Sols level)
Email	Important communications at project level

Internally, we use Discord amongst the team members as it provides us with very useful features such as dedicated channels for project, product release, channels to seek help from team members and so on. It also provides a quick way for the entire team to enter into a voice call and sort something out and also share their screen if needed.

The team has agreed that we should reply to team messages within 24 hours. This allows for the project to progress smoothly and also for the Scrum Master to understand how the team is coping with the workload.

6.0 Quality Assurance

6.1 Quality Assurance Plan

Further detailed in the Quality Assurance Plan document.

6.2 Git management - CI/CD

7.0 Definition of done

- Application functionality completed (Complete Client Goals)
    - Functionalities outlined with user stories in product backlog is completed
    - Key requirements/goals
        - Stated in QA plan document and product backlog
- Code and features passed the testing
    - Unit testing with Robot Framework
    - *Acceptance testing*
        - *Proof of concept*
    - *End-toEnd testing*
- Non-Functional requirements complete
    - UI/UX meets client requirements
- Artifacts updated
    - Project Management Plan
    - Risk Register
    - Product Backlog
- Handover documentation completed
- Application demonstrated and accepted by client

8.0 External Artifacts and Documents

The following is a list of relevant documents for the team

Artifact	Purpose
Risk register	Identifies potential risks for the project and has mitigation plans in place in case an issue arises. Refer to for any issues encountered during project
Requirements- user stories	Intended to act as a product backlog to be updated during sprint planning. Helps team organise requirements in a prioritised list
~~Team Jira board~~ e-Sols Jira board (Internal to e-sols)	Task management tool which will be used to assign tasks to team members and track progress of the team's tasks
System architecture vision document	Document which proposes an architecture development for the project and the tools and techniques which will be relevant for the project
Figma prototype (provided by clients) ECMO wireframes (internal)	Interactive User Interface mock-up of the interactive textbook. Refer to for UI design during the development stage.
Quality Assurance Plan	Document Quality Assurance process and planning
Resources and links	Document resources and links for future reference

9.0 References