Fpt University

TRƯỜNG ĐẠI HỌC FPT

CAPSTONE PROJECT REPORT

Report 2 – Project Management Plan

– Ho Chi Minh City, October 2025 –

Table of Contents

I. Record of Changes

Date	A* M, D	In charge	Change Description
28/10/2025	A	Duong Minh Nhat	Adding document
10/11/2025	M	Duong Minh Nhat	Updating document
01/12/2025	M	Duong Minh Nhat	Reviewing and updating document

*A - Added M - Modified D - Deleted

II. Project Management Plan

1. Overview

1.1 Scope & Estimation

#	WBS Item	Complexity	Est. Effort (man-days)
1	**Web Application for Admin**		**22**
1.1	FE-01: Login/Logout	Simple	2
1.2	FE-02: Manage personal profile	Simple	2
1.3	FE-03: Manage user account	Medium	4
1.4	FE-04: Manage program (assign courses)	Medium	5
1.5	FE-05: Manage course	Simple	3
1.6	FE-07: Manage class (assign instructor)	Medium	6
2	**Web Application for Simulation Manager**		**32**
2.1	FE-08 & FE-09: Auth & Profile	Simple	2
2.2	FE-10: Manage simulation component	Complex	6
2.3	FE-11: Manage simulation action	Medium	5
2.4	FE-12: Manage simulation practice	Complex	7
2.5	FE-13: Manage practice step	Complex	6
2.6	FE-14: Manage practice warning	Medium	3
2.7	FE-15: Manage simulation timeslot	Medium	3
3	**Web Application for Instructor**		**28**
3.1	FE-16 & FE-17: Auth & Profile	Simple	2
3.2	FE-18: Manage class member	Simple	2
3.3	FE-19 & FE-20: Manage sections & materials	Medium	6
3.4	FE-21: Manage quiz (assign to section)	Medium	6
3.5	FE-22: View & assign practices	Simple	3
3.6	FE-23: View results & provide feedback	Medium	5
3.7	FE-24: View overall performance & grading	Medium	4
4	**Web Application for Trainee**		**24**
4.1	FE-25 & FE-26: Auth & Profile	Simple	2
4.2	FE-27: View programs/enrollment	Medium	4

4.3	FE-28 & FE-29: View syllabus & materials	Simple	3
4.4	FE-30: Quiz attempt & history	Complex	8
4.5	FE-31: View practice results	Simple	3
4.6	**System Integration (API & Database)**	**Complex**	**4**
5	**3D Simulation Application (Desktop)**		**45**
5.1	SI-01: Login/Logout (API Integration)	Simple	3
5.2	SI-02: List practices & selection	Medium	4
5.3	SI-03 & SI-04: Guides, Settings & Steps UI	Medium	6
5.4	SI-05: Simulation Physics (Lift, Rotate, Inspect)	Complex	15
5.5	SI-06: Error detection & warning system	Complex	10
5.6	SI-07: Result calculation & submission	Medium	7
	Total Estimated Effort (man-days)		**151**

1.2 Project Objectives

#	Testing Stage	No. of Defects	% of Defect	Notes
1	Reviewing	40	35%	
4	System Test	45	40%	
5	Acceptance Test	20	25%	

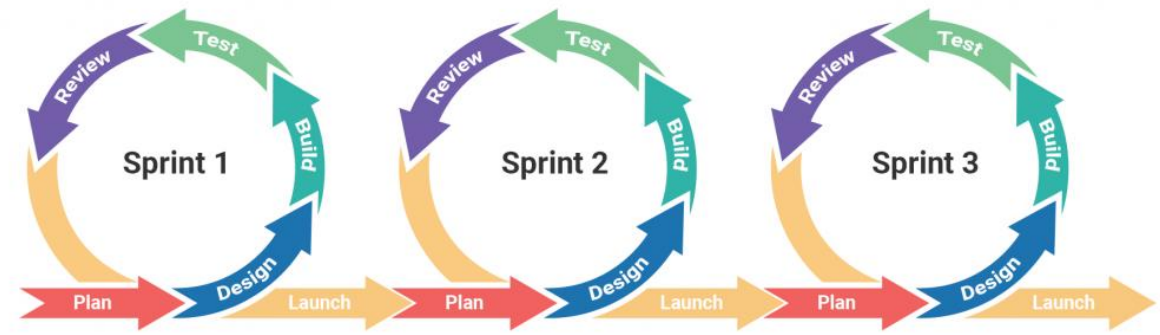
1.3 Project Risks

#	Risk Description	Impact	Possibility	Response Plans
1	Technology risk: to target available tools and framework for project overall requirements	High	Medium	Do research and testing of different technical solutions before developing main features.
2	Requirement misunderstanding risk: to correctly specify each requirement based on real-world scenarios and best-practice solution	High	High	Verify requirements with evaluation from realistic data or references, have reviews with stakeholders.

2. Management Approach

2.1 Project Process

The project will be executed using the Agile software development methodology, specifically the Scrum framework. This approach allows the team to adapt to changes quickly, deliver incremental value, and ensure continuous feedback throughout the development lifecycle.



Sprints: The development timeline is divided into 2-week Sprints. Each sprint focuses on delivering a specific set of features (from the Product Backlog) that results in a potentially shippable product increment.

Key Roles:

- Product Owner (Leader): Responsible for defining the product vision, managing the Product Backlog, and prioritizing features based on stakeholder value.
- Scrum Master (Rotational/Assigned): Ensures the team adheres to Scrum practices, removes obstacles (blockers), and facilitates meetings.
- Development Team (All Members): Cross-functional members responsible for designing, coding, testing, and integrating the software.

Scrum Ceremonies:

- Sprint Planning (Day 1 of Sprint): The team selects items from the Product Backlog to work on during the sprint and breaks them down into tasks.
- Daily Stand-up (15 mins): A brief daily meeting where members report what they did yesterday, what they will do today, and any blockers they are facing.
- Sprint Review (Last Day of Sprint): The team demonstrates the completed work to supervisors/stakeholders to gather feedback.
- Sprint Retrospective (After Review): The team discusses what went well, what didn't, and how to improve processes for the next sprint.

2.2 Quality Management

To ensure the "Learner Management and 3D Simulation System" meets high-quality standards and functional requirements, the team will implement the following quality assurance measures:

Defect Prevention:

- Coding Standards: The team will adhere to standard C# (.NET) and JavaScript (React) coding conventions to ensure code readability and maintainability.
- Static Code Analysis: Tools will be used in the IDE (Visual Studio/VS Code) to catch syntax errors and potential bugs early during development.
- Clear Requirements: All User Stories must have defined "Acceptance Criteria" before development begins to prevent misunderstanding.

Reviewing:

- Peer Code Reviews: All code changes must be submitted via Pull Requests (PR) on GitHub. At least one other team member must review the code for logic errors, style violations, and potential performance issues before it is merged into the main or develop branch.
- Document Reviews: Project documents (SRS, SDD) are reviewed by the Supervisor and team members to ensure consistency and completeness.

Testing Strategy:

- Unit Testing: Developers are responsible for writing unit tests for critical business logic, particularly for the API (Backend) services and complex Simulation algorithms.
- Integration Testing: Verifying that different modules work together correctly.
  - Web App: Testing communication between the React Frontend and .NET Backend APIs.
  - Simulation: Testing the data exchange between the Unity Desktop App and the Backend Server (e.g., login, score submission).
- System Testing (Functional Testing): The team will execute manual test cases based on the SRS to verify that the system functions as a whole and meets all user requirements (e.g., An instructor assigning a quiz, a trainee completing a simulation).
- User Acceptance Testing (UAT): Conducted near the end of the project (or Sprint Reviews) where the Supervisors/Instructors test the system to confirm it meets their business needs.

Bug Tracking:

- Defects found during testing will be logged in GitHub Issues or the project management tool (Jira/Excel) with a priority level (Critical, High, Medium, Low) and assigned to a developer for fixing.

2.3 Training Plan

Training Area	Participants	When, Duration	Waiver Criteria
Unity	All members	2 weeks	Mandatory
Blender	All members	2 weeks	Mandatory
ASP.NET	All members	2 weeks	Mandatory
React	All members	2 weeks	Mandatory
Github	All members	2 weeks	Mandatory

3. Project Deliverables

#	Deliverable	Due Date	Notes
1	Project Introduction document	24/10/2025	Overall project description
2	Project Management document	26/10/2025	Project estimation and management approach
3	SRS document	05/11/2025	Software Requirement Specification
4	Software Design Document	15/11/2025	System Design, Detailed Design, Database
5	Software Testing Document	05/12/2025	Test Document, Test Report
6	Final Report Document	10/12/2025	Project Final Report
7	Web Application	10/12/2025	Project Web Application Product
8	3D Simulation Application	10/12/2025	Project Desktop Application Product
9	Source Code Repository	10/12/2025	Complete source code for all subsystems.
10	Project Package	10/12/2025	Final codes & documents

4. Responsibility Assignments

D~Do; R~Review; S~Support; I~Informed; <blank>- Omitted

Responsibility	**NhatDNSE182236**	**DucDTSE180110**	**KhoiPDSE170462**	**QuangLNSE170415**
Project Introduction document	D	R	R	R
Project Management document	D	R	R	R
SRS Document	S	R	D	D
Software Design Document	D	S	D	R
Software Testing Document	R	S	R	D
Final Report Document	D	R	S	S
Web Application	D	R	D	D

3D Simulation Application	R	D	R	S
Source Code Repository	D	R	R	R
Project Package	R	D	D	D

5. Project Communications

Communication Item	Who/ Target	Purpose	When, Frequency	Type, Tool, Method(s)
Working with supervisor	Supervisors (Mr. Ngo Dang Ha An and Mr. Pham Thanh Tri) and team members	• Review Project requirement and documentation • Demonstrate Major features • Evaluate technologies, solutions and result	1 times per week	Face to face, Google Meet
Working in team	Team members	• Specify tasks, reviews and opinions • Ask for member's help • Report working status to leader	Always	Discord, Zalo, Face to face

6. Configuration Management

6.1 Document Management

- Using Google Drive to save documentations:
- Using Google Sheet to take notes:
- Using Draw.io to save design and diagrams:

6.2 Source Code Management

- Using Github to store source code:
  - Backend API Repository:
  - Frontend Web App Repository:
  - Simulation Desktop App Repository:
- Follow these rules when using Git:
  - Commit convention:
    - Commit code with syntax: git commit -m "type: what-commit-does"
    - Types of commits include:
      - feat: new feature
      - fix: handle bug

- config: change related to config file
- docs: changes related to config file documentation
- test: adding testing code

- Pull request convention:
  - Attach a link to the task on Notion in the description of the pull request
  - Do not push code directly to main/master branch. Let's create a new branch for every new task
  - Tag the leader for code review

6.3 Tools & Infrastructures

Category	**Tools / Infrastructure**
Technology	Unity (Simulator), Blender (Modeling), .NET (Backend), React (frontend)
Database	Microsoft SQL Server
IDEs/Editors	Visual Studio, Visual Studio Code
Diagramming	Draw.io, Lucidchart
Documentation	Ms Office, Google Docs/Sheets
Version Control	GitHub (Source Codes), Google Drive (Documents)
Deployment server	Microsoft Azure
Project management	Jira, Google Sheets