

Development Plan

SFWRENG 4G06 - Capstone Design Process

Team 17, DomainX

Awurama Nyarko
FatemeH Hamidizadeh
Fiona "Fei" J. Xie
Ghena Hatoum

Table 1: Revision History

Date	Developer(s)	Change
September 17, 2025	Fei Xie	Created first draft of document

This document outlines the development plan for the project. Covering details on Intellectual Property, team roles and expectations, workflow plan, and project scheduling. With additional sections that include the expected technologies and coding standards. This document will help ensure all members are aware of the expectations on the team, and help provide a roadmap for the project.

1 Confidential Information

This project will be open-source, no confidential information to protect.

2 IP to Protect

This software and associated documentation files for this project are protected by copywrite as dictated by the [MIT License](#).

3 Copyright License

This project use's the MIT License, located in the [LICENSE](#) file.

4 Team Meeting Plan

Meeting minutes and attendance will be taken for all meetings as their related issue in the project's Github. Each member is responsible for verifying their own attendance in the related Github issue.

4.1 Lectures

Members are expected to attend every **lecture** for Software Engineering Students **in-person** (maximum 3/week).

4.2 Tutorials

Members are expected to attend all TA-lead tutorials. For tutorials without a pre-designated lesson plan, the team will decide what to do, such as having a check-in, working session, and etc.

4.3 Team Meetings

Weekly team meetings are held every **Thursday 2:30-3:30pm virtually**. All members are expected to attend and it is up to the responsibility of the individual to inform the rest of the team why they cannot make the weekly meeting, and provide their updates asynchronously. Meetings will be structured as follows:

1. Each member will give a short recap of the work they've done since the previous meeting
2. Each member has the opportunity to bring up any blockers, concerns, or issues they are facing
3. Group will discuss and determine action items for the project that are to be done by the next meeting, and assign them to member(s).
4. Meeting minutes will be taken and recorded as a corresponding Github Issue.

4.4 Supervisor and Domain Expert Meetings

Supervisor meetings are scheduled when needed. Meeting topics must be prepared beforehand. One team member will be designated as the **main meeting coordinator** for each requested meeting. They are responsible for coordinating the meeting between the supervisor/domain expert and the rest of the team, including sending the meeting invite on Microsoft Calendar.

5 Team Communication Plan

- **Microsoft Teams:** Main way of communication between team members, both through the chat and online team meetings.
- **Github** Tracking of the project process and meeting minutes. Github issues will be created for all meetings (Lecture, Peer Review, Supervisor Meeting, etc) and code tasks.
- **Email:** Communication with the supervisor and domain expert, will all team members cc-ed.

6 Team Member Roles

Table 2: Team Roles

Name	Role(s)	Description
Awurama Nyarko	Research Lead	
Fatemeh Hamidizadeh	Backend and Database Lead	
Fei Xie	Scrum Master and Devops Lead	
Ghena Hatoum	Tech Lead	

[You should identify the types of roles you anticipate, like notetaker, leader, meeting chair, reviewer. Assigning specific people to those roles is not necessary

at this stage. In a student team the role of the individuals will likely change throughout the year. —SS]

7 Workflow Plan

The main branch of the project is protected. Meaning it cannot be directly committed into, all changes must be merged in via a pull request. Following are the general steps of what development should look like:

- Create a new **Project Work** issue, ensure the following are assigned:
 - Assignees: Who is working on the project, multiple members can be assigned
 - Label: Assign the appropriate label for the issue:
 - * feature: Adding new functionality
 - * refactor: Updating existing code without altering functionality
 - * bug: Error found in code
 - * documentation: Updating documents, no code changes
 - Milestone: Assign to the related project deliverable
 - Projects: This should be automatically set to DomainX, if the Project Work issue template was use
 - * If not, add the project label to the issue and set the project to DomainX if not done automatically by the workflow action.
- Create a branch from the issue, branch name should include the issue number.
 - By default, the issue branch should be created from main, but can also use other feature branches as the head
 - Branch naming format: **{Issue #}**-name-of-branch
- Commit changes with descriptive names
- Add tests
 - **Unit tests** must be created for feature, refactoring, and bug issues
 - **Integration tests** must be created for feature issues, optional for refactoring and bug
- Create pull request
 - All Github Actions need to partnerships
 - At least one approval from a teammate
- Merge changes into main branch

8 Project Decomposition and Scheduling

The team's Github Project, [DomainX](#), will be used to organize project-related work (non-meeting issues).

[How will the project be scheduled? This is the big picture schedule, not details. You will need to reproduce information that is in the course outline for deadlines. —SS]

9 Proof of Concept Demonstration Plan

What is the main risk, or risks, for the success of your project? What will you demonstrate during your proof of concept demonstration to convince yourself that you will be able to overcome this risk?

10 Expected Technology

[What programming language or languages do you expect to use? What external libraries? What frameworks? What technologies. Are there major components of the implementation that you expect you will implement, despite the existence of libraries that provide the required functionality. For projects with machine learning, will you use pre-trained models, or be training your own model? —SS]

[The implementation decisions can, and likely will, change over the course of the project. The initial documentation should be written in an abstract way; it should be agnostic of the implementation choices, unless the implementation choices are project constraints. However, recording our initial thoughts on implementation helps understand the challenge level and feasibility of a project. It may also help with early identification of areas where project members will need to augment their training. —SS]

Topics to discuss include the following:

- Specific programming language
- Specific libraries
- Pre-trained models
- Specific linter tool (if appropriate)
- Specific unit testing framework
- Investigation of code coverage measuring tools
- Specific plans for Continuous Integration (CI), or an explanation that CI is not being done
- Specific performance measuring tools (like Valgrind), if appropriate
- Tools you will likely be using?

[git, GitHub and GitHub projects should be part of your technology. —SS]

11 Coding Standard

[What coding standard will you adopt? —SS]

Appendix — Reflection

[Not required for CAS 741 —SS]

The purpose of reflection questions is to give you a chance to assess your own learning and that of your group as a whole, and to find ways to improve in the future. Reflection is an important part of the learning process. Reflection is also an essential component of a successful software development process.

Reflections are most interesting and useful when they're honest, even if the stories they tell are imperfect. You will be marked based on your depth of thought and analysis, and not based on the content of the reflections themselves. Thus, for full marks we encourage you to answer openly and honestly and to avoid simply writing "what you think the evaluator wants to hear."

Please answer the following questions. Some questions can be answered on the team level, but where appropriate, each team member should write their own response:

1. Why is it important to create a development plan prior to starting the project?
2. In your opinion, what are the advantages and disadvantages of using CI/CD?
3. What disagreements did your group have in this deliverable, if any, and how did you resolve them?

Appendix — Team Charter

[borrows from University of Portland Team Charter —SS]

External Goals

[What are your team’s external goals for this project? These are not the goals related to the functionality or quality of the project. These are the goals on what the team wishes to achieve with the project. Potential goals are to win a prize at the Capstone EXPO, or to have something to talk about in interviews, or to get an A+, etc. —SS]

Attendance

Expectations

[What are your team’s expectations regarding meeting attendance (being on time, leaving early, missing meetings, etc.)? —SS]

Acceptable Excuse

[What constitutes an acceptable excuse for missing a meeting or a deadline? What types of excuses will not be considered acceptable? —SS]

In Case of Emergency

[What process will team members follow if they have an emergency and cannot attend a team meeting or complete their individual work promised for a team deliverable? —SS]

Accountability and Teamwork

Quality

[What are your team’s expectations regarding the quality of team members’ preparation for team meetings and the quality of the deliverables that members bring to the team? —SS]

Attitude

[What are your team’s expectations regarding team members’ ideas, interactions with the team, cooperation, attitudes, and anything else regarding team member contributions? Do you want to introduce a code of conduct? Do you want a conflict resolution plan? Can adopt existing codes of conduct. —SS]

Stay on Track

[What methods will be used to keep the team on track? How will your team ensure that members contribute as expected to the team and that the team performs as expected? How will your team reward members who do well and manage members whose performance is below expectations? What are the consequences for someone not contributing their fair share? —SS]

[You may wish to use the project management metrics collected for the TA and instructor for this. —SS]

[You can set target metrics for attendance, commits, etc. What are the consequences if someone doesn't hit their targets? Do they need to bring the coffee to the next team meeting? Does the team need to make an appointment with their TA, or the instructor? Are there incentives for reaching targets early? —SS]

Team Building

[How will you build team cohesion (fun time, group rituals, etc.)? —SS]

Decision Making

[How will you make decisions in your group? Consensus? Vote? How will you handle disagreements? —SS]