

Development Plan

ProgName

Team #, Team Name
Student 1 name
Student 2 name
Student 3 name
Student 4 name

Table 1: Revision History

Date	Developer(s)	Change
Date1	Name(s)	Description of changes
Date2	Name(s)	Description of changes
...

[Put your introductory blurb here. Often the blurb is a brief roadmap of what is contained in the report. —SS]

[Additional information on the development plan can be found in the lecture slides. —SS]

1 Confidential Information?

[State whether your project has confidential information from industry, or not. If there is confidential information, point to the agreement you have in place. —SS]

[For most teams this section will just state that there is no confidential information to protect. —SS]

2 IP to Protect

[State whether there is IP to protect. If there is, point to the agreement. All students who are working on a project that requires an IP agreement are also required to sign the “Intellectual Property Guide Acknowledgement.” —SS]

3 Copyright License

[What copyright license is your team adopting. Point to the license in your repo. —SS]

4 Team Meeting Plan

[How often will you meet? where? —SS]

[If the meeting is a physical location (not virtual), out of an abundance of caution for safety reasons you shouldn’t put the location online —SS]

[How often will you meet with your industry advisor? when? where? —SS]

[Will meetings be virtual? At least some meetings should likely be in-person. —SS]

[How will the meetings be structured? There should be a chair for all meetings. There should be an agenda for all meetings. —SS]

5 Team Communication Plan

[Issues on GitHub should be part of your communication plan. —SS]

6 Team Member Roles

[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

- How will you be using git, including branches, pull request, etc.?
- How will you be managing issues, including template issues, issue classification, etc.?
- Use of CI/CD

8 Project Decomposition and Scheduling

- How will you be using GitHub projects?
- Include a link to your GitHub project

[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

External Goals

Our team's external goals include:

- Achieve a grade of 11 or greater in the capstone course.
- Produce a project that team members can discuss in future job interviews.
- Produce a project that team members can take pride in as a representation of 4 years of Software Engineering courses.
- Gain experience in project management and working as a cohesive team.
- Build strong, professional connections within the team, with faculty/supervisors.

Attendance

Expectations

Our team's expectations regarding attendance are as follows:

- Team members must attend all scheduled meetings, both virtual and in-person, unless a valid reason is provided in advance.
- Team members are expected to arrive on time and stay for the entire duration of the meeting. Repeated lateness or early departures without prior notice or valid reasons are unacceptable.
- Team member that are missing a meeting must notify the team at least 24 hours in advance, or as early as possible, except in the case of emergencies.
- Team members should come prepared, having completed any assigned tasks and actively participating in the productivity of the meeting.

Acceptable Excuse

Acceptable excuses for missing a meeting or arriving late:

- Health-related issues.
- Family emergencies or urgent personal matters.
- Conflicts with other courses or prior academic or professional commitments.
- Unavoidable technical difficulties.

Unacceptable excuses:

- Forgetting the meeting time or oversleeping.

- Non-urgent personal commitments (e.g., social events, non-critical errands).
- Lack of preparation for the meeting.
- Laziness.

In Case of Emergency

Our teams process for emergencies:

- The team member must inform the team as soon as possible through the groupchat explaining their situation.
- If the team member cannot attend the meeting, they should provide a brief update on their progress or any relevant information so the team can continue without them (as soon as safely and conveniently possible).
- In the case that a deadline cannot be met, the team member must alert the team and the team will confer on the urgency of the task and reassign tasks accordingly.
- For major deliverables, the team may confer to try to accommodate a reasonable extension. Team members are expected to be willing to collaborate and contribute extra work to help cover for the lost time from the emergency and put the team first.

In case of prolonged emergencies, the team will review the situation and may reassign responsibilities to ensure the project remains on track.

Accountability and Teamwork

Quality

Our teams process and expectations for quality:

- Each team member is expected to come fully prepared for meetings, having completed any assigned tasks and ideation processes.
- Deliverables and tasks should be done to the best of the team member's abilities.
- At least one other team member will have reviewed each task. If a task is below the teams accepted quality, the reviewer is expected to respectfully bring it to the attention of the team and suggest improvements.
- Team members are expected to provide constructive criticism to other team members, and team members are expected to handle the feedback professionally.

Attitude

Our teams process and expectations for attitude:

- Each team member is expected to act professionally and cooperatively.
- Team members should be expected to be willing to help out on other member's tasks, providing feedback and support (within reason).
- Team members should follow the code of conduct and conflict resolution plans from Harvard University: <https://hr.harvard.edu/staff-personnel-manual/employee-conduct>

Stay on Track

To keep the team on track, we will:

- Have weekly progress check-ins to ensure tasks are being completed as expected.
- Set specific target metrics, such as attendance, commits, and task completion.

If any team member misses a meeting without reason, they have to bring coffee at the next meeting.

Team Building

To build team cohesion, we will organize at least 2 team-building activities over the course of the Capstone course and begin meetings with a personal check-in.

Decision Making

Our team will aim for decision-making by consensus, where every member's opinion is considered. In situations where consensus cannot be reached, we will vote, with the majority decision winning. For significant disputes, we will seek input from the project TA or instructor to guide the final resolution.