

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

Currently, all the data collected from the various experiments is downloaded from the apparatus in the form of Comma-Separated Values (CSV) files. These files are then manually copied onto an existing Excel template where faulty data is deleted and the rest of the data points are labelled, filtered and then analysed. As the number of experiments and the data collected from each experiment increase in size, this model of storing and analysing data has become unsustainable.

Based on the goals outlined in section 2 of the Problem Statement and Goals document, the following key risks have been identified that must be mitigated in the early development stage to demonstrate the feasibility of this project:

- **Data Migration:** As described in section 2.1, it is critical to ensure that all existing data is migrated to a scalable and extendable database without losing integrity, labelling or structuring.
- **Performance in Querying:** As described in section 2.2, it must be ensured that the database can efficiently handle queries to avoid bottlenecks that could affect system usability.

- **Parameter Comparability:** To accomplish the analysis of the consolidated datapoints, it is essential that the database can facilitate complex inter-parameter comparisons, as defined in section 2.3.

Since the remaining goals (2.4 to 2.7) are built upon the assumption of a functioning database that allows efficient querying of the data, it must be demonstrated through the Proof of Concept (PoC) demonstration that the aforementioned risks can be mitigated and that the project can be completed on schedule.

The success of the PoC Demonstration and thus, the feasibility of the project will be determined based on the following criteria:

- Ensuring that 100% of the existing data has been migrated without loss or error to the new database.
- Demonstrating that querying of data from across multiple tables can be completed within a reasonable amount of time.
- Demonstrating that all existing inter-parameter comparisons in the Excel templates have been replicated in the database.

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

1. Why is it important to create a development plan prior to starting the project?

It is important to create a development plan prior to starting the project because it forces the team to have detailed conversations about the various aspects of the project.

Although our team spent a considerable amount of time discussing our experiences and the technologies we thought would work the best for this project (taking into account the preferences of our supervisors), creating a detailed development plan compelled us to decide on not just the technologies but also the QA factors that we had not considered discussing such as branch and commit naming conventions, coding standards and more.

Having a detailed conversation with our supervisors while coming up with a PoC Demonstration Plan allowed us to communicate our goals more precisely and set clear expectations in terms of timelines and first steps.

2. In your opinion, what are the advantages and disadvantages of using CI/CD?

So far, our team has only implemented CI for the documentation parts of the project. This means that all the checks created in GitHub Actions currently only relate to LaTeX syntax checks, pdf compilation and other such conflicts. We plan on incorporating linters and other checks in our CI process over the next couple of weeks.

The initial process of figuring out how CI can be used for our project and setting it up was painstakingly arduous. The learning curve that came with adopting the CI checks seemed completely unnecessary at times especially because the checks take on average 10-12 minutes to complete.

With that said, however, we do foresee the advantages of using CI. We have already run into a couple of issues with naming conventions of the commits and branches so having a linter in place that can flag style violations and auto-fix formatting issues to make the codebase consistent would be very useful.

3. What disagreements did your group have in this deliverable, if any, and how did you resolve them?

Our group did not face any conflicts or disputes. Most of our disagreements were related to writing styles (using third person as opposed to first person or avoiding repetition of words) and naming conventions (following

the naming convention for all the branch names and the commit messages).

To be fair, it is an exaggeration to classify these issues as ‘disagreements’ because for the naming convention issue, it was mostly the novelty of these naming conventions that caused confusion. Once Jason (who set up CI and suggested the convention style) explained how to do it in detail, we did not run into the issue again.

As for the incompatible writing styles - our team decided to review all the PRs over a 2-hour call where the writing style and formatting (indenting versus using newlines) discrepancies were brought up. All the team members were receptive to the feedback and worked together to incorporate the suggestions. In instances where we had to make decisions related to formatting, we held an informal vote and chose to proceed with the decided formatting style for all our documents to avoid confusion and maintain consistency.

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]