

# Reflection and Traceability Report on Software Engineering

Team 1, BANDwidth

Declan Young

Ben Dubois

Nathan Uy

Aidan Mariglia

April 4, 2025

## 1 Changes in Response to Feedback

### 1.1 Problem Statement, Goals and Development Plan

#### 1.1.1 TA feedback

The problem statement was revised not to include the solution and to focus on describing the problem. The goals were also refined to be the main selling point and not focus on specifics. For the development plan, team roles were localised to the team roles section, and a note was added to allow flexibility in team roles. The issue template was also linked to the Issue section of the Git workflow plan. The paragraphs were also broken down to increase readability. [Commit](#)

#### 1.1.2 Supervisor feedback

Based on discussions with our client, we decided to scratch the cloud-based solution. In addition, the technologies used were updated based on what were actually used in the project. [Commit](#)

### 1.2 SRS

#### 1.2.1 TA feedback

According to our TA's suggestion, we should prune our SRS and include only the necessary requirements. So, together with our supervisor, we discussed which requirements are essential and which ones are not. Non-essential requirements are then removed. We specified priority and dates in our project planning, mentioned the user persona in our requirements, added a traceability matrix, updated vague requirements, fixed formatting and added lead-ins and captions for tables and images. All the changes are in this [commit](#)

### 1.2.2 Supervisor feedback

Based on discussions with our supervisor, we decided which requirements were necessary and which ones were not, removing requirements that were not essential. [Commit](#)

### 1.2.3 Peer feedback

| Feedback  | Change   |
|---|--|
| Traceability Matrix ( <a href="#">issue</a> )                       | Added traceability matrix ( <a href="#">commit</a> )                       |
| Rationales ( <a href="#">issue</a> )                                | Added rationales ( <a href="#">commit</a> )                                |
| Vague Functional Requirement ( <a href="#">issue</a> )              | Added details to functional requirements ( <a href="#">commit</a> )        |
| Non-Verifiable Non-Functional Requirement ( <a href="#">issue</a> ) | Update non-functional requirements ( <a href="#">commit</a> )              |
| Lack of Details in Development Planning ( <a href="#">issue</a> )   | Added details and deadlines to development plan ( <a href="#">commit</a> ) |

## 1.3 Hazard Analysis

### 1.3.1 TA feedback

As suggested in our TA's feedback, we clarified our critical assumptions. Considerable effort was made to improve the design and format of the FMEA table. These changes can be seen here: [pull request](#).

### 1.3.2 Supervisor feedback

After discussions with our supervisor, our list of Safety and Security requirements was pruned, focusing in on the most important requirements relating to error reporting and authentication. These changes can be seen here: [pull request](#).

### 1.3.3 Peer feedback

| Feedback   | Change  |
|--|---|
| Failure cause too broad ( <a href="#">issue</a> )                  | Clarified failure cause ( <a href="#">commit</a> )  |
| Missing Assumption ( <a href="#">issue</a> )                       | Add Assumption ( <a href="#">commit</a> )   |
| Failure mode could be a cause of failure ( <a href="#">issue</a> ) | Clarify failure mode ( <a href="#">commit</a> )   |
| Missing FMEA Scenario ( <a href="#">issue</a> )                    | No change was made for this piece of feedback, as the missing FMEA Scenario was considered a misuse case rather than a failure mode.              |
| Priority Assignment in FMEA ( <a href="#">issue</a> )              | No change was made for this piece of feedback as the overall plan was to address all potential failure modes rather than only high-priority ones. |

## 1.4 Design and Design Documentation

### 1.4.1 TA feedback

According to our TA's suggestions, we changed all of the implemented-by fields of our modules to be either our program if it has implemented it or a third-party program if it was implemented this way. As well, we added more detail to some anticipated changes that lacked specificity. Lastly, we added additional detail to modules and included a new diagram of how modules interact with each other to ensure these are clearer.

### 1.4.2 Supervisor feedback

Based on discussions with our supervisor, we added a new module to the MG and MIS that is used for account verification as this was a key feature requested by our supervisor for the system. Also, we removed the admin panel module from the MG and MIS, as this was a feature that our supervisor was no longer interested in.

### 1.4.3 Peer feedback

| Feedback   | Change   |
|--|--|
| Local Function Assumption ( <a href="#">issue</a> )  | Added exception for filterData function ( <a href="#">commit</a> )                                     |
| Defining Constraints ( <a href="#">issue</a> )       | Added constraint to the leaderboard module ( <a href="#">commit</a> )                                  |
| Access Routine Semantics 1 ( <a href="#">issue</a> ) | Added exceptions for handleRegistration function ( <a href="#">commit</a> )                            |
| Ambiguous Design Logic ( <a href="#">issue</a> )     | Added new method to the submit module to validate files ( <a href="#">commit</a> )                     |
| Access Routine Semantics 2 ( <a href="#">issue</a> ) | Added information related to pending/started state to handleSubmit function ( <a href="#">commit</a> ) |

## 1.5 VnV Plan and Report

### 1.5.1 TA feedback

For the VnV Plan, we addressed the TA feedback by ensuring there was sufficient detail for some areas of the VnV Plan that did not have the necessary detail to clearly and completely describe the plan. The changes based on the TA feedback for the VnVPlan can be seen in this [commit](#).

For the VnV Report, we received our feedback on this document's due date of April 4, 2025, so therefore did not make any changes as we would not have been able to complete many of the changes, such as adding more detailed user instructions in time.

### 1.5.2 Supervisor Feedback

For both the VnV Plan and the VnV Report, no changes were made based on supervisor feedback. This is mostly due to the fact that we never went into the details of our VnV plan or VnV report during our meetings with our supervisor.

### 1.5.3 Peer feedback

| Feedback   | Change  |
|--|---|
| Network conditions not specified for response time test ( <a href="#">issue</a> )      | Specified network conditions ( <a href="#">commit</a> )   |
| Non-functional requirement test missing metric ( <a href="#">issue</a> )               | Added metric for test ( <a href="#">commit</a> )  |
| Clarify static testing plan ( <a href="#">issue</a> )                                  | Added details on static testing ( <a href="#">commit</a> )  |
| Add metric for non-functional requirement test ( <a href="#">issue</a> )               | Added metric ( <a href="#">commit</a> )   |
| Missing test pass criteria for usability tests ( <a href="#">issue</a> )               | Added expected outputs for usability tests ( <a href="#">commit</a> )   |
| Add additional tests related to invalid algorithm submissions( <a href="#">issue</a> ) | Added additional tests ( <a href="#">commit</a> )   |
| Inconsistency in terminology used ( <a href="#">issue</a> )                            | Made terminology used clear and consistent ( <a href="#">commit</a> )   |
| Increase the number of items used in sorting test ( <a href="#">issue</a> )            | Increased number of items used in sorting test ( <a href="#">commit</a> )   |
| Missing output for functional requirement test ( <a href="#">issue</a> )               | Updated output from None to 404 Error ( <a href="#">commit</a> )  |
| Increase number of testers for usability tests ( <a href="#">issue</a> )               | No change was made for this piece of feedback due to time constraints that did not allow us to conduct additional usability testing with a larger number of users.  |
| Incorrect error code ( <a href="#">issue</a> )   | No change was made for this feedback, as we felt the 404 Not Found error code was more suitable than the 403 Unauthorized error code in this case. This is because users should only be aware of the IDs of their own submissions, so if an ID of a submission is specified, it should only be attempted to be retrieved from the list of submissions of the user that is making the request. |
| A specific test scenario was not covered ( <a href="#">issue</a> )                     | No change was made for this feedback.   |

## **2 Challenge Level and Extras**

### **2.1 Challenge Level**

The challenge level for this project is general. This challenge level is justified firstly by the fact that the technologies and domain knowledge required based on the expected implementation are relatively simple, especially when considering the experience that our group has with web applications. Additionally, the problem being solved is not particularly novel, since although the problem itself isn't something that is very commonly solved, the requirements/expected solution of a web application is something that is a very common solution to problems in the software industry.

### **2.2 Extras**

The extras for this project are video walkthroughs integrated into the web application and hosting cost analysis that will compare different hosting providers.

#### **2.2.1 Walkthrough**

The video walkthrough goes through the main functionalities of the web application, namely, how to submit a model, view the results and view the leaderboard.

#### **2.2.2 Hosting Cost Analysis**

The hosting cost analysis includes the required resources for hosting based on experimental data and assumptions, as well as how much it would cost per month and per year on certain hosting providers such as AWS and Hetzner.

## **3 Design Iteration (LO11 (PrototypeIterate))**

After the development of our first revision, several areas of the system were identified as needing changes. The first major change in the design of the system was the addition of a condensed leader board view. The leader board in the first iteration contained all of the results generated from a submission. This resulted in a very crowded visual that was hard to navigate. To address this, a condensed view was added, which reduced the amount of data present and improve the ease of comparing different submissions.

The next major change introduced was the addition of model privacy. After demonstrating the first revision of the system to our supervisor, he identified that the ability to submit "private" models, would be valuable. A private model is a model where the results are only visible to the user who submitted them. The final major design change from revision 0 to revision 1 was the removal of the authentication requirement for leader board viewing. Again, after discussions with our supervisor, it was determined that requiring a user to register

an account before viewing the leader board may prevent users from taking an interest in the platform and giving it a try. As such, the requirement was removed.

## 4 Design Decisions (LO12)

Due to the tight timeline that we had, we decided to use the Django web framework to simplify the development process. Another limitation is the fact that SOC algorithms take a long time to run, so we had to separate the algorithm execution into background tasks to avoid blocking the main web application.

The major assumption we made in the design of the system was that users will not try to abuse the submission process. Individual submissions require a large amount of cpu time, meaning a single malicious user could potentially overwhelm the system, preventing other users submissions from being evaluated. There was no defensive feature designed to prevent this, as users of the system will be researchers and engineering students using the system for academic or professional purposes, so responsible use of the system is assumed.

A major system constraint is that the users will submit MATLAB code. Because of this, we had to run MATLAB through Docker containers to have a controlled environment. This ensures the code runs properly, the execution is secure, and the execution environment is isolated.

## 5 Economic Considerations (LO23)

There is a market for our product because it is specifically designed for researchers/university students who are working to find the best battery state of charge (SOC) algorithms. These users are typically involved in battery research and optimization, especially in fields like electric vehicles and energy storage systems.

Marketing the product would involve targeting academic institutions and research labs where battery SOC algorithms are the main focus. This product could be promoted through research papers or university partnerships.

A version that could be sold would likely cost around \$3000 CAD to add more functionalities, for hosting cost and any license fees. A reasonable price to charge would be between \$500-\$1000 per classroom license. To make a profit, the product would need to be sold to 6-8 classes in order to cover the costs, including the marketing costs.

Currently, there are no existing users, but occasional battery research students use the system.

## **6 Reflection on Project Management (LO24)**

### **6.1 How Does Your Project Management Compare to Your Development Plan**

No, we did not follow the team meeting plan, team communication plan, team member roles or workflow plan of our development plan very closely. Although we feel like it would have been useful to follow these plans and have more structure regarding how our group functioned, we found it difficult to do so. This is largely a result of our group having worked together in the past on different projects, leading us to fall back to old habits and tendencies (especially when facing deadlines or have big time constraints) that we had when previously working together.

In terms of technology, there was a mix between using the technology we planned on using and having/deciding to change plans.

### **6.2 What Went Well?**

In terms of processes, one thing that went very well was having consistent meetings. Whether this was meetings with our supervisor, Dr. Kollmeyer and his RA, Atjen, or just meeting as a group, we felt this helped us stay organized and aware of our project's progress.

For technology, most of the different technologies, especially those used to develop web application, such as docker, django went very well. We were quickly able to learn how to use the technologies effectively to have success while working on our project.

### **6.3 What Went Wrong?**

One major thing that went wrong was not following our workflow plan precisely. We feel that although it would have required us to invest some time in the beginning, if we had all spent the time to ensure this was followed from the beginning of the project, we would have been much more organized when the project became increasingly busy.

Another thing that went wrong related to technology was the issues we faced when trying to use Matlab with our application. Because of the requirement of licenses for Matlab, we ended up running into many issues related to obtaining a Matlab license that could be used by our hosted application.

### **6.4 What Would you Do Differently Next Time?**

Next time, we would be sure to invest time at the beginning of the project to set up better habits and practices related to the processes that we follow



while also ensuring that each team member is on the same page regarding these processes. This would keep us organized for the duration of the project and allow us to work more efficiently, especially when we become busier with work. Additionally, we would spend much more time anticipating and planning for potential risks related to technology or other critical aspects of the project. This will ensure that we have plans to manage and mitigate these risks and prevent them from having very large impacts on the project once it starts.

## 7 Reflection on Capstone

### 7.1 Which Courses Were Relevant

SFWRENG 2AA4: Introduction to Software Development helped us learn tools like GIT and latex. It also introduced us to writing documentation for our code and general design patterns and testing.

SFWRENG 3XA3: Software Project Management introduced us to the concept of the software life cycle. We also got to work on a small project that has similar deliverables as the Capstone project, but on a smaller scale. This helped us gain experience in software development from start to finish

SFWRENG 3RA3: Software Requirements and Security Considerations taught us how to write and elicit requirements. This helped us in constructing our SRS and eliciting feedback from our supervisors.

SFWRENG 3A04: Software Design III is where we learned software architecture and design patterns. This helped us in designing our code.

### 7.2 Knowledge/Skills Outside of Courses

Although a lot of the major concepts that were required to complete this project were taught in courses we have previously taken, there were more specific skills related to our exact project that we had to learn. Most of these were related to the technologies that we used to implement our project. For instance, since our project required running submissions in the background, we needed to learn how to integrate Celery, a task queuing system to handle processes in the background. Using Docker for containerization is another skill we had to learn outside of our courses.