Development Plan
Software Engineering

Team #10, Five of a Kind
Omar Abdelhamid
Daniel Maurer
Andrew Bovbel
Olivia Reich
Khalid Farag

Table 1: Revision History

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

The following document will provide an overview of the team plans for the project development process. It will establish plans to promote effective collaboration throughout the entire project as well as define roles and responsibilities to help guide successful team dynamics. It will provide insight into the plans for project execution through a workflow plan that is supported by an outline of the methods that will successfully demonstrate proof of concept as the project progresses. It also provides a high-level understanding of the technological concepts that will be explored and adopted throughout the project.

1 Confidential Information?

At this time, there is no confidential information that is relevant to note within the current project scope.

2 IP to Protect

At this time, an IP is not relevant to the current project scope.

3 Copyright License

This project is licensed under the MIT License. See the LICENSE file for details.

4 Team Meeting Plan

A basic group standard will be the organization of a group meeting that occurs a minimum of at least once a week to allow for discussion on current project progress and future planning. Given weekly variations in personal schedules, there will not be a specific date and time that the group abides by each week. Instead, the individual who takes on the role of the Meeting Chair will be in charge of organizing a date and time each week that best accommodates the schedule of each person. The nature of the meeting, regarding whether it will be in-person or virtual, will be established upon determining the date and time of the meeting. As a general principle, in-person meetings will be favoured as they typically help facilitate deeper collaboration and build better relationships amongst team members. All in-person meetings should be conducted on campus, unless a different location is agreed upon by the group. If a group member is unable to attend a certain meeting, they should notify the group through the WhatsApp group chat as soon as possible.

The individual who takes on the role of the Notetaker will document all parts of the meeting. Certain points of discussion may be established prior to the meeting, either during a previous meeting or during the period leading up to the meeting. The Notetaker will work with the Team Manager to establish an agenda of all discussion points and deadline reminders prior to the meeting. This agenda will be documented in the meeting report. During the meeting, the Meeting Chair will be responsible for ensuring all points of discussion are adequately covered. At the end of the meeting, it is the responsbility of the Notetaker to ensure the meeting has been properly documented. The meeting documentation expectations have been specified under the Notetaker description in Section 6.

Meetings with our supervisor, Dr. Onaizah, will be organized based on the need to provide more in-depth explanations on the progress of the project or to facilitate a discussion if there are many questions in which Dr. Onaizah can provide insight. There will be no set schedule for when these meetings will occur. Once the team establishes that a meeting with Dr. Onaizah is required, the team will create a list of potential dates and times that they are able to present as options for Dr. Onaizah. Unless the group decides to specifically request either an in-person or online meeting to accommodate specific group needs, the nature of the meeting will be determined based on the preference of Dr. Onaizah.

5 Team Communication Plan

All informal communication should be initiated through a WhatsApp group chat that has already been established. It is the responsibility of all group members to ensure they have access to the group chat and are actively monitoring all messages that occur through this channel of communication. The primary focus of this group chat is to facilitate easy and quick communication regarding small issues and meeting organization.

Meetings should be a regular occurrence to establish consistent and clear communication that can't be achieved solely through text communication. As established within the Team Meeting Plan, there will be a standard weekly scheduled team meeting. If a discussion that starts through the group chat seems to require additional time, effort, or broader input to consider different perspectives, the discussion may be postponed until it is able to be discussed during a meeting. This means that it can either be noted as an official point of discussion within the weekly scheduled team meeting or a separate meeting may be scheduled depending on the urgency of the discussion.

The creation of issues in the GitHub team repository will be the primary way to formally document all plans, discussions and deadlines that have been discussed. All notable information that should be known by all group members must be documented as a GitHub issue. Even if information has been stated in a different communication channel, such as a group chat or team meeting, it must still be documented in GitHub to provide consistency and ensure there are no oversights.

In the event that any communication occurs outside the WhatsApp group chat or a team meeting, it must still be noted in the GitHub team repository. Individuals should do their best to not stray from the standard communication channels unless extenuating circumstances apply or an additional standard communication channel has been established by the group throughout the project progression. This will help ensure there is consistency in communication amongst the entire group.

At a minimum, the team will provide the supervisor, Dr. Onaizah, with updates at least bi-weekly. Prior to providing an update for Dr. Onaizah, the team will establish whether a meeting with her is necessary within the next week. Meetings with Dr. Onaizah will be organized as needed. Key stakeholders will be handled in a similar manner. This includes the grad student, Kaitlyn Clancy, that created the 3D printer and works directly under Dr. Onaizah. If relevant, Kaitlyn may be cc'd onto email updates to Dr. Onaizah or separate meetings may be organized.

6 Team Member Roles

Team Manager - The individual who takes on this role will take the lead on keeping the team on track. They will closely monitor the deadlines and ensure that the team is on track with meeting the course deadlines as well as the group deadlines that have been agreed upon ahead of time.

Notetaker - During meetings, this individual will be responsible for keeping track of what was discussed and who attended the meetings. They will also be required to create a report, in the form of a GitHub issue, that is accessible to each group member at the end of a meeting. This report will include a summary of what was discussed, a meeting agenda for what has to be discussed, a review of what has changed since the previous meeting, a list of key tasks to complete prior to the next meeting and an outline of deadlines that are relevant at the current stage of the project.

Meeting Chair - This individual will be responsible for organizing meetings and ensuring that the team is following the Team Meeting Plan as specified above. They will ensure that the team is meeting at a sufficient frequency to facilitate effective collaboration both within the group and with Dr. Onaizah. They will also take the lead on team discussions regarding future meetings and do their best to reasonably accommodate the schedule of all team members when determining a date and time.

Document Reviewer - This individual will take on the responsibility of monitoring the review of all documents. They will ensure that team deadlines provide reasonable buffer for review prior to the official course due date. They will organize who reviews each section of the relevant documents being submitted as well as complete an overall final review prior to submission. This final review should include verifying the consistency, formatting, and grammar of the document. They will also work closely with the code reviewer to ensure the code and documentation properly align.

Code Reviewer - This individual will have similar responsibilities to the document reviewer but will focus on reviewing the code. This includes monitoring the integration of individual code contributions during each iteration in the development stage. They will organize milestone code check-ins to ensure that after major code contributions the functionality of previous code has not been compromised in any way and still runs seamlessly. They will also work closely with the document reviewer to ensure the code and documentation properly align.

Communication Officer - The individual who takes on this role will be responsible for handling the majority of the communication with our supervisor and key stakeholders. Specific communication responsibilities include sending emails when relevant, closely monitoring their email for replies, keeping all relevant parties in the loop regarding the progress of the project, and conveying important information from these emails to the rest of the team members.

7 Workflow Plan

Git will be utilized as the primary, centralized collaboration space for both project source code and its related documentation. Members will create new branches for each feature to be implemented, as well as per each member (eg. branches will not be shared). Commits should each contain a descriptive name of all edits. When a member is complete with their work on a given branch, they will create a pull request with appropriate labels and assignees. For instance, an acceptable assignee could be someone working on a related feature. Labels should give an estimate of how long it might take to review. Each pull request should also be linked to a given issue. Comments within pull requests will also be taken advantage of, allowing feedback to be directly linked to source code or documentation.

A milestone will be created for each deliverable (both documentation and revisions) and issues created will be associated with it. Issue templates will be used in the following ways:

- lecture: to track attendance as well as general notes/questions
- meeting(s): to record meeting minutes, attendance, and catch up members that miss the meeting
- peer review: to request another team review a given section

Additionally, for general issues, new labels will be created:

- documentation
- code
- question
- easy / hard
- bug-fix
- need help

All issues should also contain a reasonable description describing the bug / feature / section of documentation.

To incorporate continuous integration, upon each push/pull request, all tests in the test suite will be run automatically via GitHub Actions.

8 Project Decomposition and Scheduling

The team will use Github Projects as the central platform to monitor the progress of the project. This tool will help ensure that issues/tasks are organized, responsibilities are clear, and the project is on track. The link to the Github project is here: Github Project.

8.1 Github Projects

- Github Projects will be used to plan, manage, and track the workflow by creating issues and organizing them through the different stages of the project.
- Each issue will be labeled appropriately to help organize them and provide clarity on the progress of the project.
- Issues will be assigned to team members to clearly identify ownership and responsibility.
- Issues will be linked to different project stages to show how tasks move through the development cycle
- Issues on the project will be linked to the different deliverables of the project to help track the progress of the project.

8.2 Project Scheduling

The project will be scheduled based on the capstone deliverables, the team will prioritize the deliverables and schedule them accordingly. At each deliverable, the sections will be broken down into smaller tasks and these tasks will be assigned to team members. The project will be scheduled to be completed in a timely manner, and to be completed to a high quality. Below is a table of the project deliverables and their corresponding due dates.

Deliverable	Due Date
Problem Statement, Proof of Concept, and Development Plan	September 22th, 2025
SRS and Hazard Analysis Revision 0	October 6th, 2025
V&V Plan Revision 0	October 27th, 2025
Design Document Revision 1	November 10th, 2025
Proof of Concept Demonstration	November 17th, 2025 - Nov 26th, 2025
Design Document Revision 0	January 19th, 2026
Revision 0 Demonstration	February 2nd, 2026 - Feb 11th, 2026
V&V Report and Extras Revision 0	March 9th, 2026
Final Demonstration (Revision 1)	March 23rd, 2026 - March 26th, 2026
Final Documentation (Revision 1)	April 6th, 2026
EXPO Demonstration	TBD

9 Proof of Concept Demonstration Plan

Our POC Demonstration Plan will begin with a pre-existing CAD file. Taking this CAD file, we will then:

1. **Show the process of importing the file into our software.** This would include converting the file properly and showing the key feature of 'slicing' the model into voxels.

2. **Demonstrate that the software can load and render models of varying resolutions.**

3. **Demonstrate that each voxel/group of voxels can be magnetized.**

4. **Export the altered file into the existing 3D printing software.**

There are two primary risks associated with this demo:

1. **Rendering the model in 3D may be slow.** Initial iterations may be laggy due to lack of group experience with 3D rendering. High memory usage may result from inefficient handling of large voxel grids, reducing responsiveness. Should this be the case, the group will need to revisit optimization strategies to find ways of improving memory usage and responsiveness.

2. **File converting time is longer than converting the model manually.** While stakeholders have stated that process automation is the ultimate goal, the fact remains that our software should be reasonably more efficient than the current solution which requires re-building the model by hand. If the software that is built does not provide a more efficient process, our group will need to re-analyse how the files are converted from

the CAD file into our software. This may require further investigation into different methods of handling the initial files that will potentially speed up the process, or reconsidering if the scope regarding voxel customisability should be paired back for initial iterations.

10 Expected Technology

The implementation of this project will be primarily web-based, with a browser-accessible frontend and a local server backend with local saves on the client machine. The frontend will be developed using **TypeScript** and **React**, with **Redux Toolkit** for state management. For 3D voxel visualization and interaction, the project expects to use **Three.js**, integrated with React for rendering and manipulation.

On the backend, the project anticipates using **Python** with **FastAPI** to provide REST endpoints for validating and normalizing voxel data, as well as managing persistence. A local instance of **PostgreSQL** is expected for structured storage of voxel geometries, materials, and related metadata.

For code quality, the frontend will adopt **ESLint** as a linter, with **Jest** for unit and integration testing. The backend will rely on **pytest** for automated testing, with a tentative coverage goal of approximately 80%. Code coverage reporting will be handled with tools such as **Istanbul**.

Version control and project management will be conducted through **git**, **GitHub**, and **GitHub Projects**, which will serve as the central platforms for collaboration, task tracking, and issue management. Continuous integration will be performed using **GitHub Actions**, with automated checks for linting, tests, and coverage included in the pipeline.

/sectionCoding Standard

To ensure consistency, maintainability, and readability, the project will adopt established coding standards for all areas of development.

For the **frontend** (TypeScript/React):

- Variable and function names will use **camelCase**, while **PascalCase** will be used for React components.
- The principle of **DRY** (Don't Repeat Yourself) will guide code organization, promoting modular components and reusable utility functions.
- File and folder structures will be organized in a consistent **kebab-case** format for directories and filenames.

For the **backend** (Python/FastAPI):

- Code will follow the **PEP 8** style guide
- Variable and function names will use **snake_case**, while classes will be written in **PascalCase**.

Across both frontend and backend:

- All code will be documented through inline comments and docstrings, particularly for APIs.
- **Unit tests** will accompany critical modules, following a goal of 80% code coverage.
- Commit messages will be written clearly and consistently, referencing related issues when applicable.
- Team members will adhere to software engineering principles such as **single responsibility**, **modularity**, and **clarity over cleverness**.

These conventions aim to produce code that is readable, maintainable, and collaborative.

Appendix — Reflection

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?

Creating a development plan prior to starting a project is essential because it provides a clear, structured roadmap for the team to follow, ensuring that all members are on the same page regarding the project's scope and requirements. This well-structured plan helps identify any potential risks or challenges early on, making it easier to address them before they become major issues. Overall, a development plan helps reduce uncertainty and improves overall efficiency by having a clear plan to follow.

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

In our opinion, the main advantage of using CI/CD when working on a project is that enables faster and more reliable development by automating builds, testing, and deployments. As a team, this helps us to catch any issues early on in the development process, and maintain a high quality of code. However, a disadvantage of CI/CD is that it requires a more complex setup and configuration, which can be time-consuming and require additional resources. In most cases, it is worth implementing CI/CD to improve the efficiency and quality of the development process.

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

In this deliverable, the group did not have any disagreements. We were able to complete the deliverable in a timely manner and with a high quality of work.

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 fo 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

To stay on track as a team, we will follow these necessary methods:

1. **Regular Team Meetings:** We will hold meetings at least once a week to discuss progress, address any issues, and plan for the next deliverable. This meeting will be communicated through the team chat *WhatsApp*. This meeting will be attended by all team members, unless otherwise specified.

2. **Task Management:** We will use GitHub Projects to assign tasks, track progress, and ensure everyone is contributing their fair share. This is crucial to ensure that all team members contribute their fair share to the project.

3. **Regular Updates:** We will update the supervisor and stakeholders regularly to keep them informed about our progress. This will be done through emails to the supervisor and the stakeholders. The supervisor will be updated at least once every two weeks, unless any questions arise.

For team contribution and performance, we will follow these methods:

1. **Attendance:** We will use GitHub Projects to track the attendance of each team member for team meetings, lectures and supervisor meetings. This is important to ensure that all team members are contributing their fair share to the project.

2. **Contributions:** We will use GitHub Projects to track the contributions of each team member to the project. Each team member will be responsible for contributing to this capstone project to the best of their ability. Issues will be evaluated based on the quality of the work, the deadline and the effort put into the issue.

3. **Code Review:** Each team member will be responsible for reviewing the code of the other team members to ensure that the work is completed at a high quality and on time. Each pull request will be reviewed by at least two other team members before it is merged into the main branch.

4. **Performance Metrics:** We will use commits, meetings attended, and issues completed to evaluate the performance of each team member.

For the rewards and consequences, we will follow these methods:

1. **Rewards:** To reward members who do well and to encourage good performance, we will recognize their contributions and celebrate their achievements. They will have the option to decide which task they would like to work on for the next deliverable.

2. **Managing underperformers:** We will consider a team member to be underperforming if they are not contributing to the project or are not meeting deadlines. To manage this, we will have a team meeting discussing the issue and offering support and guidance to the team member. If the issue persists, the team member will need to contribute to more tasks to make up for the work of the other team members.

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]