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
...	...	...

[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

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

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). 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.

A milestone will be created for each deliverable (both documentation and revisions) and issgit ues 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

- 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

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**. Continuous integration will be performed using **GitHub Actions**, with automated checks for linting, tests, and coverage included in the pipeline.

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.

11 Coding Standard

[What coding standard will you adopt? —SS]

4

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?

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 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

[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]