

# Problem Statement and Goals

## 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
Sept. 16, 2025	Khalid Farag	Extras
Sept. 18, 2025	Omar Abdelhamid	Problem Statement, Goals, Stretch Goals

## 1 Problem Statement

### 1.1 Problem

There is currently no CAD workflow that lets users take a finished 3D design and select individual voxels (note: a voxel is a cube building block in a 3D printer) by layers from bottom to top to assign magnetization. Without voxel selection and per-layer editing, researchers are forced to rebuild the object voxel-by-voxel (e.g., in COMSOL), manually choosing a magnetization direction for each voxel. This consumes hours per part and makes iteration fragile: if a mistake is made, there is no practical way to reselect or change a prior voxel's magnetization without redoing subsequent work. Our system removes this bottleneck by enabling per-layer voxel viewing, selection (single or multiple voxels), and re-assignment of magnetization directions on top of an imported design.

### 1.2 Inputs and Outputs

**Inputs (Day 1).**

- Mesh CAD file exported from common tools (e.g., AutoCAD) and ingested as a mesh for voxelization (e.g., STL/OBJ).
- Voxel grid initially fixed at  $300 \times 300 \mu\text{m}$  (XY) and  $110 \mu\text{m}$  (Z) to form cubes at the stated dimensions.
- User interaction via a UI that slices the object into layers (bottom→top) and allows selecting one or multiple voxels within a layer to assign magnetization.

#### Outputs (Day 1).

- A Java-readable export containing per-voxel data for the custom printer pipeline (voxel location/layer and magnetization direction metadata).

### 1.3 Stakeholders

Stakeholder	Role / Interest
Researchers / Engineers	Primary users who design and print magnetized objects; need fast, reliable voxel-level magnetization after CAD.
Lab Operators / Printer Owners	Consume the exported file to run the custom printer workflow; care about correctness and repeatability.
Supervisors / PIs (Approvers)	Validate that a magnetization plan is correct for the experiment's goals.

### 1.4 Environment

- **Platforms:** Desktop/laptop; Windows with WSL or native Linux.
- **Runtime/Integration:** Reads mesh CAD files for voxelization; exports a Java-readable file to drive a custom printer program.
- **Assumptions:** Commodity hardware (laptop/desktop) sufficient for interactive layer viewing and voxel selection.

## 2 Goals

1. **Import & Voxelization:** Import a mesh CAD file and convert it to a voxel grid with fixed voxel size of  $300 \times 300 \mu\text{m}$  (XY) and  $110 \mu\text{m}$  (Z).
2. **Layered UI & Selection:** Provide a UI to browse layers from bottom to top and select voxels (single or multiple) within a layer.
3. **Magnetization Assignment & Editing:** Assign a magnetization direction to selected voxels and allow re-selection and changes without restarting the design (with undo/redo).

4. **Export:** Generate a Java-readable file capturing per-voxel magnetization for the custom printer software.

### 3 Stretch Goals

- **Adjustable Voxel Size:** Make voxel dimensions configurable to match different printers (beyond the fixed initial values).
- **Solid CAD Ingest (Future):** Add support for solid CAD (e.g., STEP/IGES) with robust tessellation before voxelization.
- **Additional Export Formats:** Support exporting to formats beyond the Java-based output to integrate with other software/hardware.

### 4 Extras

The extras for this project are:

- **User Manual:** This manual will provide a comprehensive guide and explanations on how to effectively use the software, emphasizing the key features that differentiate it from a normal CAD software. This is useful for helping users to operate the system independently, reducing the need for direct support, and ensuring they can utilize the system to its full potential with all of the features. To make it more user-friendly, the manual will be written in a way that is easy to understand and follow, with clear instructions and examples. It will also include any safety warnings that are relevant to the software.
- **Usability Report:** This report will document the findings from usability testing, including identified pain points, user feedback, and recommendations from stakeholders for improving the user experience. It will also include information about how the testing was held, including the number of participants, the tasks they were given, the questions they were asked, and its initial results. This is useful for ensuring the product is intuitive, efficient, and satisfying for its target audience, leading to higher user adoption and satisfaction.

## 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. What went well while writing this deliverable?
2. What pain points did you experience during this deliverable, and how did you resolve them?
3. How did you and your team adjust the scope of your goals to ensure they are suitable for a Capstone project (not overly ambitious but also of appropriate complexity for a senior design project)?