Hazard Analysis
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
October 10, 2025	All	Created initial revision of Hazard Analysis

Contents

[You are free to modify this template. —SS]

1 Introduction

The following sections aim to provide insight into our project and the role that hazard analysis plays throughout its development. It introduces the concept of a hazard and how it may manifest in software development. It then connects those ideas back to the scope of our current project.

1.1 Problem Statement

Recently, master's students at McMaster University have been developing cutting-edge 3D printer technology that has the capabilities of incorporating complex elements of magnetization and multi-material components. However, with this innovative new design comes the challenge of limited software capabilities that are unable to support the unique needs of property assignment while simultaneously allowing an efficient and intuitive design. Currently, developers of the 3D printer are forced to design their model outside of COMSOL, which is the current software that permits magnetization assignment. To build their model for printing, they must build their entire model voxel-by-voxel in COMSOL, individually assigning each voxel a specific magnetization. This leads to an unintuitive, time-consuming process that is unideal as they look to further test and develop their printer. They require new software that can take a model built through AutoCAD and provide a replica that has been sliced into voxels with the ability to easily assign magnetization and material values to each voxel.

1.2 Hazard Analysis Introduction

As defined by the Canadian Centre for Occupational Health and Safety, a hazard can be defined as any source of potential for harm, damage, or adverse effect to people, property, or the environment [1]. Many of the typical factors that can contribute to a hazard, including human error, unsafe working conditions, equipment malfunction, environmental influences, and inadequate safety procedures. However, from the perspective of a software developer, hazards can extend outside the factors listed above and may take form in data corruption, unstable system components, poor development processes, security risks and technological limitations.

As described above, our project aims to create software that supports slicing a pre-existing model into voxels and permitting easy property assignment for future printing processes.This project has the potential to greatly reduce the time, energy and workload exerted on those using the multi-material 3D printer. However, to accomplish this, there are many potential hazards that must be mitigated during system development. These hazards may come from impractical technical implementations, incompatible input data structures with pre-existing software or faulty output generation that could result in unsafe or unintended behaviour in the physical system. Therefore, the hazard analysis outlined below aims to identify and assess all hazards to ensure that our team is able to successfully deploy a safe and reliable software system.

2 Scope and Purpose of Hazard Analysis

[You should say what **loss** could be incurred because of the hazards. —SS]

3 System Boundaries and Components

As discussed in the SRS, the system consists of four primary modules, each responsible for a specific part of the transformation from the initial CAD file to the completed voxel file. All components operate locally, with dependencies limited to the library files needed to interact with necessary files.

3.1 Import Module

- **Purpose:** Converts input external files into internal native type and interprets existing native files.
- **Key Functions:**
  - Parse CAD files (as per F211)
  - Generate output voxel files for program use (as per F213, F214, F215, NF211)
- **Integration:** Indirectly receives CAD files from external CAD software.

3.2 Visualization Module

- **Purpose:** Renders interactible 3D images for model visualization.
- **Key Functions:**
  - Generate an interactible 3D visualization of the voxel model (as per F221, F223, NF221, NF222)
  - Allow users to view layers of the voxel model in isolation (as per F224)
  - Visualize voxels selected by the user (as per F225)

3.3 Editing Module

- **Purpose:** Tracks and stores user edits to voxel properties.
- **Key Functions:**
  - Set properties for sets of selected voxels (as per F231, F233, F235)
  - Save per-voxel properties within project files (as per F236, NF232)

3.4 Export Module

- **Purpose:** Handles saving and exporting of user files.
- **Key Functions:**
  - Export voxel models as a native project file (as per F243)
  - Export project file as printer-ready format (as per F242, NF241)
  - Validate that each voxel has properties assigned correctly (as per F241)
- **Integration:** Indirectly integrates with printer software through export files.

4 Critical Assumptions

[These assumptions that are made about the software or system. You should minimize the number of assumptions that remove potential hazards. For instance, you could assume a part will never fail, but it is generally better to include this potential failure mode. —SS]

5 Failure Mode and Effect Analysis

In this section, we will be analyzing the failure modes and effects of the system. The following table will break down the potential failure modes, their causes, effects, recommended actions, and the safety requirements that are associated with them.

Table 2: Failure Mode and Effect Analysis (FMEA) - Part 1

Component	**Failure Mode**	**Causes of Failure(s)**	**Effects of Failure(s)**	**Recommended Action(s)**	**SR**	**Ref.**
Import Module	The system does not process the CAD file correctly as the user is expecting.	• The inputed STL file is corrupted therefore it can't be read by the system. • The input file is not in the correct format so the system can't process it.	• The system can't process the file therefore it can't be sliced. • Due to the confusion in system not reading the file, this will result in user frustration. • It will lead to user spending more time to get the system to process the file.	When the file is first inputted into the system, the system will validate that the file is first in the correct format, and that it does not contain any corrupt information. This will then lead to the system prompting the user to re-enter the file or input a new one in.	F211, NF211, SCR1, SCR9	H1
Visualization Module	The system fails to render the 3D image of the model	• The device that is being used to run the system does not have enough memory to render the image. • The device's GPU driver is not working correctly or is not up to date as required for the system to render the image.	• Model is not displayed leading to poor user experience. • Designers can't proceed with the design review, and fall back to using their old design software to add magnetization and material properties to the design.	The system should check if the device has enough memory to render the image, and if the GPU driver is working correctly. If not, the system should prompt the user to use a different device or update the GPU driver.	F221, NF221, SCR2, SCR9	H2
Visualization Module	The voxels displayed as selected/currently editing do not match the voxels actually being edited.	• Logical error in the system's selection/editing mechanism. • UI rendering bug not accurately reflecting the backend state. • Inconsistent data synchronization between front-end display and back-end data model.	• Poor user experience (UX) as the user thinks they're doing one thing, but the system is actually editing something completely different. • Incorrect modifications to the model. • Loss of user trust in the system's accuracy. • Time wasted correcting errors.	The system shall ensure real-time synchronization between the UI display of selected/edited voxels and the backend data model.	F225, F227, F228, NF221, SCR3, SCR8	H3

Table 3: Failure Mode and Effect Analysis (FMEA) - Part 2

Component	**Failure Mode**	**Causes of Failure(s)**	**Effects of Failure(s)**	**Recommended Action(s)**	**SR**	**Ref.**
Export Module and Editing Module	Exported file is missing magnetization and material properties	• The system does not export the magnetization and material properties to the metadata file correctly. • The system fails to save the magnetization and material properties for certain voxels	• The custom 3D printer software that reads the metadata file receives incomplete data causing print failure or defects. • The custom 3D printer software reads the file correctly, however the printer crashes due to missing magnetization and material properties. • Material will be wasted if printer crashes after printing a few layers.	The system should validate that the magnetization and material properties are saved for all voxels. If not, the system should prompt the user to re-export the file.	F231, F233, NF232, F241, F242, SCR4	H4
Export Module	File format is exported in the incorrect format	• The system does not export the metadata file in the correct structure and format.	The custom 3D printer software can't read the file correctly because of the wrong format therefore causing the program to crash.	The system should check that the structure of the metadata file is correct and matches the required format.	F241, F242, NF241, SCR5	H5
Full System	User progress is lost	• Unexpected software bug that causes the system to crash, which causes the system to lose its state. • The user experiences a hardware malfunction (e.g., power loss, memory failure), which causes the system to lose its state.	• User loses signifcant work and would need to re-do the work. • User will start getting frustrated and lose trust in the system. • User will lose signicant time in the total workflow of their design due to the need to re-do the work.	The system shall implement a auto-saving mechanism to periodically save the state of the user's process to allow for recovery in case of a system shutdown or failure.	F236, NF241, SCR6, SCR7	H6

6 Safety and Security Requirements

In this section, we will be analyzing the safety requirements of the system. The following table will break down the potential safety requirements, their associated hazards, and their priority.

Symbol	Value
OPTIMAL_VOXEL_SIZE	5.5nm
MAX_VOXELS	13,996,800,000 voxels
MAX_LAYER_DISPLAY	518,400 voxels
MAX_DISPLAY	103,680,000 voxels
MAX_VOXEL_SELECTION	1000 voxels
MIN_RATE	500,000 voxels per second
MAX_EDIT_LATENCY	1 second
MAX_VIEW_LATENCY	500ms
MAX_INTERACTIONS	5
INPUT_FILE_VALIDATION_THRESH	100%
MIN_RENDER_MEMORY	16 GB
UI_SYNC_ACCURACY	100%
EXPORT_COMPLETENESS_THRESH	100%
METADATA_FORMAT_VALIDATION	100%
AUTO_SAVE_INTERVAL	5 minutes
UNDO_HISTORY_SIZE	10 actions
FILE_IMPORT_EXPORT_TIMEOUT	2 minutes
FEEDBACK_UPDATE_TIME	30 seconds

Table 4: Expanded Table of Symbolic Constants. This includes constants from SRS S.2, with the addition of new constants derived during the hazard analysis process.

SCR 1. *The system shall validate the imported CAD files for their format correctness before processing the model.*
Rationale: The corrupted files can cause the system to fail and prevent successful voxelization to the workflow disruption.
Fit Criterion: `INPUT_FILE_VALIDATION_THRESH` of the imported files must pass the validation checks, and the system should reject any files that fail to do so.
Associated Hazards: H1
Priority: High

SCR 2. *The system shall check if the device has enough memory and GPU capabilities before attempting to render 3D models and voxel grids.*
Rationale: Insufficient memory or outdated GPU drivers can cause rendering failures with the system, preventing users from visualizing their models and proceeding with property assignments.

Fit Criterion: The hardware where the system is running on must have at least `MIN_RENDER_MEMORY` of memory, and the GPU driver must be compatible with the system.
Associated Hazards: H2
Priority: Medium

SCR 3. *The system shall maintain real-time synchronization between the UI component of selected/edited voxels and the backend data model.*
Rationale: Any failures in this synchronization can lead to the user editing unintended voxels, causing incorrect property assignment and loss of user trust in system accuracy.
Fit Criterion: `UI_SYNC_ACCURACY` of voxel selections and edits must be accurately reflected in both UI display and backend data model.
Associated Hazards: H3
Priority: High

SCR 4. *The system shall validate all magnetization and material properties before exporting the file containing the metadata for all voxels.*
Rationale: Missing these important information can cause excessive time and material waste, and in some cases can cause printer failures.
Fit Criterion: `EXPORT_COMPLETENESS_THRESH` of exported files must contain complete magnetization and material property data for all voxels.
Associated Hazards: H4
Priority: High

SCR 5. *The system shall validate the metadata file format and structure before exporting the file, ensuring that it is compatible with the next system that will read the file.*
Rationale: Incorrect metadata format can cause the custom 3D printer program to crash, preventing the user from continuing with the workflow, leading to user frustration.
Fit Criterion: `METADATA_FORMAT_VALIDATION` of exported metadata files must match the required structure and format expected by the next system that will read the file.
Associated Hazards: H5
Priority: High

SCR 6. *The system shall implement auto-saving functionality to preserve user progress and enable recovery from any system crashes or hardware failures.*
Rationale: Any system crashes or hardware failures can lead to signicant amount of work loss, which would require the user to restart the workflow from scratch, which would be very time consuming and frustrating.
Fit Criterion: The system must automatically save user progress every `AUTO_SAVE_INTERVAL`.
Associated Hazards: H6
Priority: High

SCR 7. *The system shall provide undo/redo functionality to allow users to correct their mistakes without losing signifcant work.*
Rationale: Human errors in design are common and users need the ability to easily correct these mistakes without restarting the process.
Fit Criterion: The system must maintain an undo history of at least `UNDO_HISTORY_SIZE` actions and provide clear undo/redo controls.
Associated Hazards: H6
Priority: Medium

SCR 8. *The system shall limit the number of voxels that can be selected simultaneously to prevent any performance issues.*
Rationale: Selecting too many voxels at a time can cause system slowdowns, UI freezing and an overall poor user experience.
Fit Criterion: The system must limit voxel selection to a maximum of `MAX_VOXEL_SELECTION` voxels at any time and provide clear feedback when limits are reached.
Associated Hazards: H3
Priority: Medium

SCR 9. *The system shall provide progress updates and timeout handling for long running operations such as importing/exporting the file, and voxelization.*
Rationale: Large CAD files can take significant time to process, and users need feedback on progress to avoid waiting indefinitely.
Fit Criterion: All import and export operations must timeout after `FILE_IMPORT_EXPORT_TIMEOUT`, and all operations exceeding `FEEDBACK_UPDATE_TIME` must provide progress updates.
Associated Hazards: H1, H2
Priority: Medium

7 Roadmap

In this section, we will state which safety requirements will be implemented as part of the capstone timeline, and which requirements will be implemented in the future.

Requirements that will be implemented as part of the capstone timeline, as they are considered to be high priority, and will be implemented first:

- SCR 1
- SCR 2
- SCR 3
- SCR 4
- SCR 5

Requirements that will be implemented in the future, this also contains requirements that are considered to be high priority, and will only be implemented if time permits:

- SCR 6
- SCR 7
- SCR 8

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. What went well while writing this deliverable?

Collaboration was the key word with this deliverable. With how this document was intrinsically tied to pieces of the SRS (e.g. components, functional requirements), we each had to ensure our parts not only remained consistent across both documents, but who finished what when was also a concern. Thankfully, this was able to be sorted out. To ensure all members were on the same page about what was to be discussed, initial brainstorms from one member were received and implemented by another, whose work was finally judged by the first member. Suggestions of possible hazards were also taken not just from the group, but from discussions with the client, who verified our concerns. Despite our project not exhibiting traditional safety concerns (beyond the remote possibility of printer damage), possible hazards were nevertheless identified as they relate to user productivity, an essential facet of project success.

2. What pain points did you experience during this deliverable, and how did you resolve them?

Omar:

Daniel: Responsible for writing the System Boundaries, my main concern was finding a succinct way of explaining each component without simply restating what was already discussed in the SRS. Providing the necessary detail for the HA whilst also summarizing was an interesting balance to strike. Ultimately I settled on leaving out items not discussed within the analysis itself. In addition, when brainstorming possible hazards, tracing back possible causes when the system obviously isn't developed was a challenge. In the end, we realised that discussing causes more generally was sufficient in indicating a possible cause.

Andrew:

Olivia:

Khalid: I was primarly responsible on the Failure Mode and Effects Analysis (FMEA) for this deliverable. This involved identify any potentail failure modes with the system that we have not thought about during the requirements gathering process. This included analyzing their causes and

effects, and assessing their severity and likelihood. The challenge that I had during this process was identifying any safety requirements or hazards. This is because our system is a software tool that doesn't deal with any physical harm to a user, so we had to think more abstractly about the hazards. The hazards that I came up where more directly towards user productivity, data corruption, loss of work, or connectivity issues with the external systems.

3. Which of your listed risks had your team thought of before this deliverable, and which did you think of while doing this deliverable? For the latter ones (ones you thought of while doing the Hazard Analysis), how did they come about?

The major risks already identified were (which were refined in this analysis):

1. System not processing the file correctly. This was considered during work on the development plan, as it is the starting point of the workflow.

2. Performance degradation from selecting too many voxels simultaneously. This risk was identified during the development plan as a potential risk, and was refined in this analysis.

3. System not providing an undo/redo functionality to combat the risk of unintended edits being made to the model. This is a risk that we identified after meeting with the supervisor and discussing the system's requirements.

4. System not saving user progress. This is a risk that was thought of during the development plan, as we discussed how the system should handle user progress by autosaving.

5. System validating the material and magnetization properties before exporting the file. This is a risk that was thought of during the development plan, when we discussed the requirements for the system with the supervisor.

The risks that were thought of while doing this deliverable:

1. The device does not have enough memory or GPU capabilities to render 3D models and voxel grids. We came up with this risk after researching how much 3D imaging requires memory and how the GPU driver can affect the rendering process.

2. The system is not maintaining real-time synchronization between the UI and backend data model. We came up with this risk after noticing how this could lead to unintended edits being made to the model and cause the backend data to potentially become corrupted.

3. Long-running operations causing user frustration due to lack of progress feedback. We came up with risk after learning about feedback in the SFRWENG 4HC3 course, where we learned how the Norman Principle of Feedback is important for user experience.

4. System not exporting the file in the correct format. We came up with this risk after exploring how the Java program that reads the metadata file can crash if the file is not in the correct format.

4. Other than the risk of physical harm (some projects may not have any appreciable risks of this form), list at least 2 other types of risk in software products. Why are they important to consider?

The terms listed here are general suggestions as areas of risk in software products.

1. "Usability harm." This includes any hazard that might harm the user's experience, satisfaction, or productivity when using the product, and can arise from faulty user interfaces or common errors that the user must find workarounds for. This is important to consider as if the product's usability is a substantial decrease versus another solution, users will be encouraged to find alternate products that accomplish their goals without these pain points. In the context of our project, many of the risks identified relate to this concept (e.g. progress loss).

2. "Virtual harm." Whereas physical harm relates to humans being harmed, virtual harm involves harm to files, programs, and posssibly hardware. This is important to consider as files can be just as precious as human safety, especially in the case where no backups exist. This can also factor into usability harm; if these errors are common, users can completely lose trust in the product. In our project, this manifests as both incorrect edits being made to user files, as well as complete progress loss.

References

[1] C. C. for O. H. and S. Government of Canada, "CCOHS: Hazard and Risk - Hazard Identification." Accessed: Oct. 08, 2025. [Online]. Available: https://www.ccohs.ca/oshanswers/hsprograms/hazard/hazards_identification.html