

Development Plan ScoreGen

Team #7, Tune Goons
Emily Perica
Ian Algenio
Jackson Lippert
Mark Kogan

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

4.1 General Team Meetings

Meeting Chair: Jackson Lippert.

Frequency: Weekly.

Format: In-person.

Location: McMaster’s main campus.

- Particular location may vary, will be discussed prior to the meeting time.

Time (primary): Monday, 2:30pm-4:20pm (the designated tutorial time slot for SFWRENG 4G06A/B).

Time (secondary): Monday, 12:30pm-1:30pm.

- The secondary time slot will be used if the primary slot is occupied by synchronous tutorials or at the discretion of team members.

Structure:

| |
|---------------------------------------|
| AGENDA ITEMS (GENERAL MEETING) |
|---------------------------------------|

- | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none">1. Discuss/review team member updates.<ol style="list-style-type: none">(a) Recent changes made.(b) Problems encountered.(c) Status of previous meeting action items.2. Go over new business.<ol style="list-style-type: none">(a) Discuss future deliverables/tasks to be done.(b) Define individual and/or group next steps.3. Assign action items. |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

4.2 Supervisor/Industry Advisor Meetings

Frequency: Biweekly.

Format: In-person, virtual if necessary.

Location: Dr. Martin v. Mohrenschildt's campus office.

Time: By appointment.

Structure:

| AGENDA ITEMS (SUPERVISOR MEETING) |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none">1. Discuss/review team updates.<ol style="list-style-type: none">(a) Project milestones reached.(b) Status of previous meeting action items.2. Troubleshoot recent issues that require supervisor-specific knowledge.3. Determine high-level plan for project progression.<ol style="list-style-type: none">(a) Discuss problems the team expects to encounter in the near future.(b) Identify goals the team wishes to achieve by the next meeting with the supervisor.4. Assign action items (if applicable).5. Discuss details of next supervisor meeting.<ol style="list-style-type: none">(a) Date & time.(b) Location. |

Note: All fields are subject to change depending on the supervisor's availability or the team's need to consult the supervisor.

5 Team Communication Plan

5.1 General Purpose Communication

The primary mode of communication for general purposes will be through Apple iMessage SMS/MMS in the team group chat. In the event that a team member does not respond via iMessage, the secondary mode of communication will be used. This will be through an Instagram direct messages group chat. The primary and secondary communication channels were chosen due to their practicality, team members will almost certainly have their smartphone on hand, which they can use for both iMessage and Instagram direct messages. A tertiary form of communication through Microsoft Teams direct messaging and/or video calls. This will be used in the event a team member does not respond within a reasonable time frame to primary and secondary communication methods. It will also be the main application through which virtual meetings will take place (both general and supervisor meetings) as it has screen sharing capabilities and is familiar to all team members.

5.2 Implementation and Task Assignment

GitHub Projects will be the primary tool used for project management related communication. A kanban board will be collectively maintained by the team for task assignment, progress status, meeting bookkeeping, and item reviews. GitHub Issues will be used in tandem with the kanban board. As such, every opened issue will be required to also appear on the kanban board. In addition to project tasks and to-dos, issues will also be opened for every team meeting and every SFWRENG 4G06A/B lecture to track attendance and team contribution. There are five categories of issues that can be opened by the team, each having their own template:

- Lecture
- Meeting
 - General team meetings
 - Supervisor meetings
 - TA meetings
- Peer Review

The use of these templates standardize issue creation and streamlines project progression.

6 Team Member Roles

All listed team member roles are tentative and subject to change at the discretion of the team. Should role changes occur team members should follow the process outlined below:

1. Contact all team members that are relevant to the role change.
2. Discuss proposed changes and ensure each team member understands their new responsibilities.
3. Notify uncontacted team members of the change and/or the overtaking of responsibilities.

6.1 Project Manager

Assignee: Emily Perica

The project manager directs the team towards successful completion of the project. They will be responsible for coordinating the progression by creating meeting agendas for the meeting chairperson to follow. They will also maintain the Kanban board and create and assign GitHub issues that concern the entire team.

6.2 Meeting Chair

Assignee(s): Jackson Lippert

The meeting chair is responsible for driving team meetings towards productivity and efficiency. This will be done by guiding and directing meetings to minimize off-topic conversation. They are required to ensure all meeting agenda items are addressed in a timely manner. Finally, they will facilitate constructive, collaborative discussions during meetings.

6.3 Meeting Minutes Recorder*

Assignee(s): Ian Algenio

The primary responsibility of this role is to document and record team meetings. They will also track team member attendance and address associated issues. They will record meeting discussions, issues, topics, action items, etc.

6.4 Subject Matter Expert*

Assignee(s): Mark Kogan

The subject matter expert is given the responsibility of maintaining a neat and organized record of source materials used during the lifetime of the project. This may include academic articles, open-source repositories, external libraries used, etc. They will also create any necessary citations for project documentation elements.

6.5 Software Developer*

Assignee(s): ALL MEMBERS

All team members will take on the responsibilities of this role. As developers they are responsible for designing, testing, coding, and maintaining the software components of the project.

6.6 Quality Assurance and Testing Specialist*

Assignee(s): ALL MEMBERS

This role ensures the software meets stakeholder needs and quality standards. They will develop and execute test plans and test cases, measure code coverage, and report defects in the software. Additionally, they will verify the software meets functional and non-functional requirements such as performance, security, usability, etc.

6.7 Music Domain Expert

Assignee(s): Emily Perica

As the expert in the music theory domain, this person will address any confusion regarding general music theory such as terminology. They will share their prior knowledge and experience with music whether that's playing an instrument or

clarifying details about the domain.

Note: Any team member roles marked with an asterisk (*) are able to be assigned to multiple people, have overlapping responsibilities with other roles, and/or have responsibilities are likely to be shared amongst the entire team.

7 Workflow Plan

Issue Management Workflows

When creating an issue in the Capstone Kanban Board, there are two possible workflows:

1. Meeting issue is created (using the relevant template) prior to a scheduled meeting.
 - (a) Issue is given the 'meeting' label and assigned the 'Meeting Minutes' status.
 - (b) The notetaker (see section 6) will add a comment to the issue, outlining the meeting minutes and action items of each attendee.
 - (c) Issue is closed once all action items have been addressed.
2. Project work issue is created using a blank issue.
 - (a) Issue begins in 'To Do'
 - (b) One or more appropriate labels are assigned - 'documentation', 'deliverable', 'enhancement', 'bug', 'DevOps', and potentially others as the need arises.
 - (c) If the issue does not have the 'deliverable' label AND is not time sensitive, it will be moved to 'Backlog'.
 - i. The issue will move out of 'Backlog' and into 'To Do' if the issue becomes time sensitive or the team has the time/resources to take it on.
 - (d) The issue moves to 'In Progress' once it is assigned and being worked on. It may be assigned to more than one person, depending on the scope of the issue.
 - i. At this point, an issue pertaining to the codebase will branch to the Git Workflow (below).
 - (e) The issue moves to 'In Review' once a PR has been made and approval is requested from the team.
 - (f) The issue moves to 'Done' and can be closed once all associated tasks have been completed, including all related PRs being merged/closed.

In addition to the above usage of the Kanban board, all course deliverables are documented as milestones in the repository and will have issues assigned as

necessary. Each milestone has a due date and a description of the tasks to be completed.

The Project Manager (see section 6 for team member roles) is responsible for making issues and keeping the project on track. As well, they will run monthly backlog grooming sessions with the team to ensure any stale issues are taken care of.

Git/GitHub Workflow

A new developer on the project will begin at step 1; otherwise step 1 can be skipped:

1. Developer may choose to fork or clone the repo. A fork is recommended so that contributors can display the project directly in their GitHub profile.
2. Main is a protected branch - any commits must be made in a separate branch before being pushed to a PR. There is no specific naming scheme for branches.
3. Commit and PR names will both start with 'Issue #: '. This will be enforced through a GitHub Action.
 - (a) Each PR will only address a single issue, but an issue may be spread out across multiple PRs.
 - (b) Squash commits are enforced to ensure there is a single commit per PR.
4. At least one reviewer must approve the PR before it can be merged into main.

CI/CD Workflow

Actions on Push/Pull Request:

- LaTeX to PDF converter
- Code packaged as an executable
- Linting
- Unit tests
- PR naming convention enforcer

Note that a PR may not be merged if any failures are present in the pipeline.

8 Project Decomposition and Scheduling

Link to our GitHub Project - [Capstone Kanban Board](#).

| Due Date | Deliverable |
|-----------------|-----------------------------------------------|
| Sept 24/24 | Problem Statement, POC Plan, Development Plan |
| Oct 9/24 | Requirements Document Revision 0 |
| Oct 23/24 | Hazard Analysis |
| Nov 1/24 | V&V Plan Revision 0 |
| Nov 11/24* | Proof of Concept Demonstration |
| Jan 15/25 | Design Document Revision 0 |
| Feb 3/25* | Revision 0 Demonstration |
| Mar 7/25 | V&V Report Revision 0 |
| Mar 24/25* | Final Demonstration (Revision 1) |
| Apr ##/25 - TBD | Expo Demonstration |
| Apr 2/25 | Final Documentation (Revision 1) |

*The demonstration should be prepared by this date, but this will not necessarily be the date of the demo itself.

Ideally, progress on a deliverable will begin at least 3 days before the previous deliverable is due (i.e., D2 work should begin on or before Sept 21). The above schedule will be enforced through the use of Milestones in the project repository, such that each deliverable (including its due date and all relevant information) is its own Milestone. The only exception to this is the Expo Demonstration, which is not expected to require any specific issues.

Each deliverable will be split into issues based on dependencies and relevant tasks. For example, when writing this document a single issue consisted of sections 7 and 8. They both cover project management-related plans, so completing one section provides the writer with the needed knowledge to write the other section. An alternative proposed method was to create an issue for each section of the document, but the granularity of this strategy may make the document feel more disjointed to the reader.

9 Proof of Concept Demonstration Plan

The Proof of Concept will be a demonstration of the team’s ability to develop the most crucial aspect of the final product, which is to take in a single note and output the note pitch and length. If successfully performed, it will clearly show that a viable product can be produced given enough time.

The main risk associated with the success of the project is potential nuance introduced by human performance on instruments. Determining the time signature and differentiating between similar pitches such as $A\sharp$ and $B\flat$ could be potentially difficult tasks. It will become necessary to test and benchmark accuracy during audio edge cases.

The Proof of Concept will help dispel concerns by allowing the team to show the product’s potential for accurate signal processing.

Given the ability to accurately process audio, the completion of the product will be simplified, as the focus can shift towards construction of a user interface and adding features while optimizing performance and accuracy.

10 Expected Technology

10.1 Tabular Overview

| DOMAIN | TOOL/TECHNOLOGY | DESCRIPTION |
|-----------------------|-----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Version Control | Git, GitHub | For version control and issue tracking respectively. |
| Project Management | GitHub Projects | Used for Kanban board, task assignment, backlogging, etc. |
| Back-End Development | C/C++, MATLAB | Possible specific languages, high performance and speed is required for signal processing. |
| Sound File Conversion | libsndfile | A specific library that was researched and will likely be used for sampled sound file format conversion. |
| Code Formatting | Linters | Specific tools are undetermined at this stage as it is language dependent. To be used with CI workflows to format code. |
| Testing | MATLAB unittest/automation, CMake, Unity Test, etc. | Specific tools are undetermined, likely a unit testing framework that does not transcend languages (i.e. language-specific). |
| AI/ML | N/A | Will use existing libraries (pytorch, tensorflow, etc.) given the time constraint of the project, however, the team aims to generate our own datasets for training. |
| Front-End Development | HTML, CSS, JS, React Native | Languages and existing libraries deemed useful for web-based GUI. A framework will likely be used, in particular React Native. |
| UI/UX | Figma, Canva | To design and effectively prototype UI/UX elements. |
| Hardware | Audio Interface | An external unit for I/O of audio signals to the host device. Possible to use the host computer's built-in microphone as opposed to this. |
| Musical Instruments | Keyboard, guitar, ukelele, etc. | Used to generate input signals. |

10.2 Continuous Integration Plans

GitHub Actions (GHA) will be used for continuous integration. They will employ automated workflows for various unit and integration tests as well as code formatting/linting. One specific workflow that will be created is .tex to .pdf format conversion to ease documentation editing. GHA will also ideally be used to test code coverage metrics. One of the major components of the project is the signal processing algorithm. Depending on the implementation details, different metrics in addition to the basic ones such as statement coverage might be used. For example, the designed algorithm might benefit from condition or path coverage for pitch detection.

11 Coding Standard

[What coding standard will you adopt? —SS]

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?

Emily: The development plan that we've created in this document will serve as the groundwork that the rest of the project will be built off of. We've already seen what a difference a formalized plan can make compared with jumping in blindly - our first ideas for this project have already changed and been refined significantly since organizing our thoughts and anticipating our future needs. In creating this development plan we've been able to set realistic goals and discuss within the group each of our individual visions. One of the benefits of working in teams is combining different people's thought processes and background knowledge, and yet this may also provide a disadvantage when communication is lacking. By creating a solid plan ahead of writing any code, it allows us to ensure that our levels of communication are up to the standards expected from a final year capstone project.

Ian:

Jackson:

Mark:

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

Emily: CI/CD acts as the last barrier before deployment, and ensures that only well-tested code makes it to production. In our case, it will help us to maintain a uniformity across our codebase and prevent us from collecting tech debt as the project matures. That being said, it also will require a significant amount of effort to get running smoothly, both from the development side and the planning/project management side. It can also be resource intensive, which may be hard for a group of students to handle and maintain.

Ian:

Jackson:

Mark:

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 for 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]