Development Methodology

We will be developing with a scrum-based agile methodology, with two week sprints, with sprints starting and ending on Tuesday.

Meetings

We will have a sprint planning meeting at the Thursday team meeting right before the beginning of the next sprint to determine what work we will get done in the upcoming sprint based on the work we got done in the previous sprint. We will have an asynchronous standup in the standup channel in the discord. Every Monday, Wednesday and Friday each team member should send a message to the standup channel outlining their status, what they plan to do between then and the next standup, and any issues they're running into. We will also have sprint retrospectives at the end of every sprint to review how the team has been working together and what we need to change about our process.

Tools for work tracking

We will be using a github projects board to track the progress of work items from the project scope, each item will be a separate ticket on the board attached to an issue in the GitHub repository. Each PR (discussed more below) will be linked to the github issue, so it is easy to track what work is being done on the issue. Items will be moved from the icebox to the backlog when they are ready to be worked on, and at the beginning of each sprint we will move items that we plan on working on in the sprint from the backlog to the todo column, and team members will assign themselves to items.

Tools for quality assurance

To ensure quality we will use a code review process. When a team member starts development on a user story they will create a new branch in the github repository, and when the work is done and manually tested the team member will put up a pull request to merge the code into the master branch. At least 1 review should be required to merge code in. Reviewers are expected to review the code for quality, and test the changes locally on their machine to ensure that there are not cross-platform issues. On large changes, the frontend or backend lead (depending on where the changes are) should also review the code to make sure it is architecturally cohesive with the rest of the code in that section of the project.