

Table 1: Revision History

Date	Developer(s)	Change
Date1	Name(s)	Description of changes
Date2	Name(s)	Description of changes
...

Development Plan

Farming Matters

Team #14, The Farmers

Brandon Duong
Andrew Balmakund
Mihail Serafimovski
Mohammad Harun
Namit Chopra

[Put your introductory blurb here. —SS]

1 Team Meeting Plan

The team members will meet at least twice a week at flexible times, either in person or online. Formal meetings with the supervisor and members will occur every Wednesday at 10:45 am. The meeting scribe will be responsible for recording the contents and topics discussed at each meeting. Other team members will create a list of activities and topics to discuss as well as time estimates for each item. All attendees will have a chance to contribute to the discussion and have their input taken into account. Group decisions and work will be part of a team-only discussion. Individual tasks will be assigned during these meetings to be completed before the next meeting. The meeting scribe will post the contents summary of the meeting in the Facebook chat which is available to all team members.

All members must be present for all meetings. If a member is absent, they must notify the rest of the group one hour before the meeting with a reason. If the meeting cannot be postponed, the absent member must reach out to inquire about the content discussed in the meeting.

2 Team Communication Plan

The team will have a Facebook Messenger group to communicate any administrative and miscellaneous issues with each other. Facebook Messenger will be the main form of communication. It will be used to set up meeting times, and discuss small ideas about different aspects of the project, and documentation issues. For code-related issues, the team will track and communicate these tasks

using GitHub issues. The team will be using E-mail to communicate with the professor and supervisor.

3 Team Member Roles

4 Workflow Plan

For the Git workflow, we will use [trunk-based development](#). The main branch will be protected such that no update can be pushed to it directly. This will ensure that the main branch will always be stable. A new and separate branch will be created for every change that is based on the main branch. A pull request must be created and approved before it can be merged into the main branch. All tests should pass and at least one developer, other than the author or the pull request collaborators, must review the code before approving the pull request. The feature branch will be deleted once the feature branch is merged into the main branch.

Each task will be added as a GitHub issue and classified using the default labels including bug, documentation, and feature. Developers will assign themselves issues based on priority and familiarity. Once a developer's task is finished and merged into the main branch, the corresponding issue will be closed. A template will be set up on GitHub issues for a standard bug report and feature request.

5 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?

6 Technology

- Specific programming language
- Specific linter tool (if appropriate)
- Specific unit testing framework
- Investigation of code coverage measuring tools
- Specific plans for Continuous Integration (CI), or an explanation that CI is not being done
- Specific performance measuring tools (like Valgrind), if appropriate
- Libraries you will likely be using?
- Tools you will likely be using?

7 Coding Standard

8 Project Scheduling

[How will the project be scheduled? —SS]