



**Diverse Makers**

# **Team Reflection**

12-13-2024

**Project Sponsor: Dr. Jared Duval**

**Faculty Member: Isaac Shaffer**

**Project Mentor: Vahid Nikoonejad Fard**

**Team members:**

**Daniel Minichetti (Team Lead)**

**Kane Davidson**

**Eduardo De La Rosa**

**Elleana Negrelli**

**Aaron Ramirez**

## **Overview**

In this document, we outline our team's reflection on the Diverse Makers project, examining our software development process, tools utilized, and team dynamics throughout the course of developing our accessible STEM education mobile application.

**Overview:** At the end of a project, it's useful to go back and reflect on how the project went, how the team functioned, how effectively you used tools, and so on. This worksheet is designed to guide you in this process, and capture the outcomes.

**How to fill this out:** Hold a final team meeting, after you've turned in the last deliverable and the heat is off. Order a pizza, crack open a beverage. Then sit down as a team and go through the following worksheet, discussing and filling in each section. Type up the result, and email the document to your team mentor.

**Grading Metrics:** You will not be graded on the *content* of this document per se. That is, if for instance, your self-assessment concludes that you "didn't use version control tools effectively", then this shortcoming won't affect your grade; the point is that it should be an honest assessment. What you *will* be graded on is *how well* you fill in this document: thoughtful self-analysis gets a perfect score; cursory/lame/vague self-analysis will score low. We instructors use this document to help us think about how to encourage more learning and better teaming on projects, so please help us out!

---

**Team Name:** Diverse Makers

**Team members:** Daniel Minichetti, Kane Davidson, Eduardo De La Rosa, Elleana Negrelli, and Aaron Ramirez

Course number and name: CS486, Senior Capstone Design

Semester: Fall 2024 Date this reflection completed: 12/09/2024

## Software DESIGN PROCESS

**How did your team structure** the software development process? Did you choose a particular formal model (SCRUM, Agile, etc.). If so, which one and why? If not, did you explicitly agree on an informal process...or was it just pretty random. Explain briefly.

**The overall approach that team members used in the software development process was an agile model where often, sprint sessions were conducted to accomplish blocks of the project at a time. Given the relatively short time frame to build our project it was personally what worked best for the entire team to ensure key milestones and deadlines were met, while leaving adequate time for debugging and iterating.**

**How did it go?** Now briefly discuss how satisfied you were with this process. Did it work well for this project? Why or why not?

**Given the scope of this project I believe our method was efficient and successful in meeting all components of the project. Given that the main goal was to lay a foundation of sorts for a crowd-sourced app, having short bursts of development on individual components of the app were straightforward and ended up saving us a lot of time later in the semester.**

**What changes might you make** in your development process if you have it to do again? More structure? Less? Different process model?

**I think a more robust outline of who wanted to work on what would've been preferred. It was at times redundant to assign multiple people to one or so given tasks so having a more established rubric of who was working on what component of our project would've likely increased efficiency more. This issue didn't end up being too much of a problem given the rather quick development window we had set up anyways but it's a hindsight that could've further increased our productivity.**

## Software DEVELOPMENT TOOLS

**What software tools or aids**, if any, did your team members use to support or organize software development? For each of the following categories, list the tool(s) used, and briefly describe how the tool was actually used. If you didn't use a formal tool, explain how you handled the matter with informal means.

- Source creation tools: IDEs, text editors, plugins, anything used to edit/create source.

**Visual Studio Code was our primary IDE for development.**

- Version control: How did you manage your codebase?

**GitHub was the primary method for managing the version control of our app. A majority of major updates were documented on our GitHub repository to keep track of what changes were made from version to version.**

- Bug tracking: How did you keep track of bugs, who was working on them, and their status

**A mix of our GitHub and our Discord chat group were utilized to manage bugs and assign who would work on fixing specific ones. This kept formal and informal communication on what bugs existed and how they would be resolved by team members.**

- UML modelers and other miscellaneous tools:

**Draw.io served as our main UML modeler to create architectural diagrams.**

**How did it go?** Comment on any problems or issues related to organizing the coding process. How might you have managed this better? Were some tools you used superfluous or overkill? What tools or mechanisms would you try next time to deal with those issues better?

**I think some issues we had starting out early in development was getting used to the inner workings of GitHub and effectively updating our repository because of that fact. None of the tools utilized per say were overkill but definitely had a learning curve for some of the team members making it difficult to keep up with the rapid development of the project. It's unlikely that more efficient means exist to deal with this sort of issue and it was more or less just a learning process of using GitHub to consistently keep track of any developments, bugs, etc. Having a consistent communication channel through Discord helped tremendously with communicating these issues and getting assistance from other team members.**

## **TEAMING and PROJECT MANAGEMENT**

Without getting caught up in detailed problems or individual blame, take a moment to think about how your team dynamics worked overall. Here are a few questions to guide you:

**How did you organize your team?** Did you have some clear distribution of team roles (leader, technical lead, documentation lead, etc.) up front? Or was it more just “everyone does everything as needed”?

**I believe from the start we as a team had established our roles and responsibilities, our team in a vacuum worked really well together without much overhead besides just basic reminders. I believe even though there were proper roles established with set responsibilities, everyone was willing throughout the semester to take on other responsibilities as necessary to adapt to situations or problems as they arose making team organization easy. Everyone was very flexible and that was a key trait to making the project development as smooth as it was.**

**How did you communicate within the team?** Comment on each of the following communication mechanisms:

- Regular team meetings? If so, how often?

**Team meetings occurred every week at a scheduled time.**

- Impromptu team meetings? If so, roughly what percent of total team meetings were of this sort?

**Roughly 20% or so of our team meetings outside of our weekly meetings were impromptu, this was usually to address any urgent or unexpected issues so that they were on everyone's radar ahead of time.**

- Emails to all members? If so, explain briefly: about how often, what used for?

**Email was rarely used as a communication tool outside of communicating with our client and team mentor, as we primarily used our Discord channel to communicate. Any emails directed towards communication with our team mentor or client had everyone CC'd on the email to keep all team members in the loop.**

- Software tools? Were any of the software tools you mentioned above (e.g. bug/issue tracking) used to communicate and organize tasks, e.g., in lieu of emails or other discussion?

**Since GitHub and Discord were our primary communication channels for any sort of bug/issue tracking, all of our task assignments were conducted in Discord as it was the easiest for everyone.**

- Other communication channels used? Facebook, wiki, text messages, phone conferences, etc.

**Discord was primarily used to communicate and organize tasks, it simply was the most efficient communication channel for everybody and made addressing challenges, issues, or any other occurrences a breeze to communicate and resolve. Organizing tasks, scheduling meetings, and communicating amongst the team in general was most efficient on this platform.**

**How did it go?** Did you feel that intra-team communication overall went well? Were there breakdowns, e.g., where someone didn't know something was due, didn't realize a task had been assigned to him/her, did not know about a deadline, etc.? Without getting into details, simply comment on whether such breakdowns occurred, what the overall cause was, and how serious (if at all) the consequences were.

**Our team had very good communication on all fronts and I would say that really no breakdowns actually occurred throughout the semester. I believe we had a very disciplined communication schedule that allowed us to work ahead on a lot of due dates and avoid missing any deadlines. In events where deadlines were coming up we all collectively made the effort to find a time to buckle down and get the work done, again even if it meant taking up responsibilities not exactly established in our respective roles. This was pivotal not only in our day to day communication but our project success as a whole.**

**What could you do better?** More structured leadership? A more formal task assignment/tracking system? Using better/other communication mechanisms? Generally just

think about what you all would do next time to improve communication and avoid breakdowns mentioned.

**Some improvements could've been made to setting up the respective roles for the actual development of our project. The actual coding part of the project felt a little less structured than our other capstone work outside of the app itself but it is again just another small optimization that could've been made to increase team efficiency and productivity.**

**Nice work! Congratulations on finishing your project! Please enter all of your answers in this electronic document and send it off to your instructor or team mentor.**

### Some closing thoughts...

Spend a little more time on your own percolating on the answers you gave in this self-reflection exercise. Being effective as a project team is *not easy* (!!), and is a skill that we all have to work on continuously. There is rarely any single or simple reason why a project was a bumpy ride; usually it's a combination of factors...of which is YOU. Regardless of project or team, there are things that could have been done differently to make it flow better. Recognizing those things through thoughtful reflection post-facto is the key to improvement!