

NAU-CS Team Project Self-Reflection Worksheet

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: _____ FairyMander _____

Team members: Izaac Molina, Sophia Ingram, Ceanna Jarrett, Dylan Franco,
Jeysen Angous _____

Course number and name: CS486C Capstone Experience _____

Semester: Fall 2024 _____ Date this reflection completed: 12/10/2024 _____

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.

Our team essentially used a SCRUM process, where we had weekly "standups" and used the weekly task report as a tracker for all of our progress. This allowed us to keep track of our progress on each task and indicate the outcomes for each tasks

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?

It worked out in the end, but we could have done a better job regularly updating the task report. Most of the time, our team waited until the report was formally due in order to update the report. To help remedy some of this, we created a more informal channel in our discord for updating each other on milestones. This helped each other understand the progress being made on the project and allowed us to ask for feedback

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

The team could have directly adopted a more formal organizational structure such as a more formal SCRUM or Agile methodology. We could have also had a more fine grained task report, as most of the time the tasks we assigned were very broad and general.

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.
 - VS Code, Jupyter Notebook, Sublime Text
- Version control: How did you manage your codebase?
 - GitHub
- Bug tracking: How did you keep track of bugs, who was working on them, and their status
 - No formal bug tracking, just assigned bugs in task report and addressed them as needed
- UML modelers and other miscellaneous tools:
 - Draw.io, Excel

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?

For the most part, these technologies were appropriate, however, we initially wanted to utilize github's code review features, but ended up doing things in a much more "scrappy" way. If we had more formal code review, we likely would have run into less communication problems and the team as a whole would likely have an improved understanding of the codebase overall.

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"?

Each team member had a distinct role in the project:

- Jeysen was the website manager and designer, ensuring everything was responsive and kept up to date.
- Ceanna was mostly involved with working on our Fairness module. She was the recorder for the team, and worked on generating some of the states.
- Dylan was heavily involved in data cleaning and obtaining the data necessary for our algorithm to work. He was also responsible for generating some of the districts.
- Sophia specialized in generating the districts and finding the best ways to use our algorithm effectively. She communicated back and forth with the client.
- Izaac was the main developer for the python package, and also acted as the team lead by managing our tasks

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

- Regular team meetings? If so, how often?

Discord once weekly, once weekly with our mentor.

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

Very rarely, if ever. Most issues were able to be resolved via instant message.

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

Every email sent to our mentor, sponsor, or professor was also CC'd to all teammates.

- 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?

No, most tracking was done in the team discord and in the weekly task reports

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

The team communicated via discord, CC'd emails, and Zoom meetings.

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 two major "road bumps" along the way.

The first was our team encountering problems/issues and not having a clear way to communicate these problems. To remedy this, we created the updates channel in our discord for posting clarifying questions and/or asking for feedback.

The next was making sure each team member was taking initiative in the project development process. We would often miss details with certain assignments, which was improved by making a commitment as a team to have each member involved in understanding the requirements for each due date throughout the project and communicating when these requirements were not being met.

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.

As mentioned previously, a lack of a formal code review process led to issues in our code. Having a formal review process would have likely led to fewer misunderstandings and would have improved each team member's understanding of the project as a whole.

We also could have done more preemptive research before we began coding. We knew that due to the theoretical nature of our project, a high amount of research was necessary, but even still, we certainly could have done more research into the feasibility of our fairness methods and chosen technologies, as these changed from our initial feasibility report.

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!