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 and 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: In Your Orbit

Team members: Austin Carlile, Noah Schwartz, Nicholas Gonzalez, Minuka Trikawalagoda

Course number and name: CS486C Computer Science Capstone

Semester: Spring 2025
Date this reflection completed: 29 April 2025

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 did use a sort of SCRUM methodology for our capstone project. While we did not do daily standups, we did 3 separate weekly standups, one for the team, one with the mentor, and one with the client. The Fall Semester was comprised of weekly meetings with both the team and mentor but the Spring Semester included weekly meetings with the client as well. As per the SCRUM methodology, we did sprints for each deliverable cycle, broke up tasks at our

weekly team meetings, and had demos or presentations of deliverables at every meeting.

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?

We believe this process worked quite well for our project development lifecycle. It felt like we could all contribute equally and stay on task for each sprint especially since our client required a demo of some sort every week to insure we continued making progress.

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

We agreed the only thing we would change significantly would be the length of sprints as there were some weeks we had to push back client meetings for the sake of creating and presenting better demos which made the team lack some confidence at times even though we were making progress.

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.
- Version control: How did you manage your codebase?
- Bug tracking: How did you keep track of bugs, who was working on them, and their status
- UML modelers and other miscellaneous tools:

Source creation tools varied between the team members but in general VS Code and Notepad++ were the main development environments along with WSL to run the project environment. Our codebase was managed on Github that all members were able to contribute to. As far as bug tracking, it was more oral through our communication channels in Discord and when we saw each other in our Mobile Development classes T/Th. For UML modeling, we employed Draw.io as we were all familiar with the site through coursework.

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?

We easily agreed that the biggest coding process challenge was setting up the specific environment necessary for running the USGS tools (ALE and SPICE). First off, all of us use the Windows operating system which does not run the tools well so figuring out either WSL or dual booting Linux was difficult. Then, once we had a Linux environment, we had to struggle through the scarce documentation on the USGS tool setup. One thing we agree we would try to do better is document our individual progress with the coding process so we would not need to start from square one on each machine.

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"?

We were very grateful that Austin stepped up and delivered so well as Team Lead. While we all did work on the project, it was clear that Austin kept us on track and was willing to take on any extra work to meet our deadlines whenever a team member needed some additional help. While we did initially assign roles to each team member, those roles kind of blurred which made it a bit harder to divide up and complete some tasks. It was overall a solid team effort and everyone contributed what they could when possible.

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

- Regular team meetings? If so, how often?
- Impromptu team meetings? If so, roughly what percent of total team meetings were of this sort?
- Emails to all members? If so, explain briefly: about how often, what used for?
- Software tools? Were any of the software tools you mentioned above (e.g. bug/issue tracking) using to communicate and organize tasks, e.g., in lieu of emails or other discussion?
- Other communication channels used? Facebook, wiki, text messages, phone conferences, etc.

The team utilized the Discord application for communication both within the team and with the client for quick questions and discussing any changes to meeting times. The team met in-person, unless something required a Zoom meeting or cancellation, once a week, three times, once as a team, once with the mentor, and once with the client. Emails were used for communication with the mentor outside of meetings if any questions arose concerning deliverables.

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.

We felt overall that intra-team communication went quite well. All team members were very responsive and there were minimal communication faults, especially in the second semester. If a communication fault occurred, it was quickly remedied with very little drawback to the development process. Communication with the client was a bit difficult and slow at first but the team fixed this error quite early on by creating a separate channel in the team Discord to ensure quick and efficient communication when needed.

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.

In general we feel the team did quite well across the entire process, however next time we think a more formal and used task tracker would have helped a bit with completing some tasks on time as sometimes, even with the weekly spreadsheet task tracker, some tasks were a little mixed between members. Additionally, we had decided on specific roles for each member at the beginning of last semester but then sort of blurred those lines as the process developed which may have led to this problem.

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

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!