

# 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:** Team Mockingbird

**Team members:** Austin Malmin, Charles Saluski, ShanHong (Kyle) Mo, Conrad Murphy

Course number and name: CS 486 Capstone Experience

Semester: Fall 2022      Date this reflection completed: 12/5/22

## 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 not explicitly write software instead we simply combined different libraries to make the overall product work. We did meet with our sponsor once a week so there was some agile components

**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 well for the most part. With this project, only one person could really work on it at a time rendering the other members useless so one person would sprint then take a break til their next "shift" with the hardware

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

The way we did it was great but maybe a better definition of what a "shift" looked like would have slightly improved it.

## 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.
  - Most of our code was prewritten for us and just downloaded and stored inside the file system of our pi. We did not really use any tools except catkin make and ROS which hardly count as software tools.
- Version control: How did you manage your codebase?
  - N/A
- Bug tracking: How did you keep track of bugs, who was working on them, and their status
  - Since only one person could work at a time, whoever had the hardware was in charge of documenting what they did, in the install guide and linking any fixes they found in the guide as well. As most of our code was already written and tested it was largely bug free but when an error occurred from incompatibility, it took googling the error msg to find clean solutions.
- UML modelers and other miscellaneous tools:
  - Diagrams.net

**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?

Overall it sucked that only one person could work at a time but there is not an easy solution to this problem. We really used the bare minimum tools to bring the project together which kept our system lightweight. This enabled it to run on a lackluster Raspberry Pi.

## 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”?

Aside from team lead, task reports, and final edits, everything was split equally. Austin largely communicated with Dr. Shenkin, the mentors and Dr. Leverington. Conrad performed many of the final edits and stylistic choices after the paper was freed of grammatical errors and such. Kyle was the sole creator of the task reports.

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

- Regular team meetings? If so, how often?
  - Met once a week on Mondays with the following:
    - Sponsor 10am -11am
    - Internal meeting 11am -12pm
    - Mentor 12pm-until it is done
- Impromptu team meetings? If so, roughly what percent of total team meetings were of this sort?
  - We did not have many impromptu meetings except during weeks of presentations to practice. The time of this meeting was agreed upon during the Monday internal meeting
- Emails to all members? If so, explain briefly: about how often, what used for?
  - Rarely used as members of the team did not regularly check email. We are focused on using Discord as it is a great platform to inform all members and it also allows meetings online.
- 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, as only one person could work at a time for the hardware, If we had any issues with it or require team assistance we will use Discord to communicate.

- Other communication channels used? Facebook, wiki, text messages, phone conferences, etc.
  - Discord was our main source of communication internally as well as with our client. Discord offered us the opportunity to create channels for separate tasks, chat logs to store images and files, voice channels for quick voice calls, etc...

**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.

Overall, there were no major breakdowns in internal communication, however there was some breakdown in communication between our team, the posted assignments and the mentors. As we had a change in mentors mid semester and a change from virtual to in person classes, some assignments and expectations were not clear. When these questions were raised to our mentor, he also had just started this role and was also not sure on the best way to move forward. This however was of little consequence as when there was a breakdown grace was shown to every party involved

**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.

Austin – Better organized task assignment than a list on the whiteboard which was sent into the Discord. While effective, it was not very formal and could have been confusing to people not in attendance at the meeting.

Conrad – Our task organization could have been better. When we divided parts of the paper assignments between team members, it was done on the whiteboard once and then sometimes forgotten about. Occasionally tasks were written down in Discord as well, but it wasn't very well organized. The task reports didn't help with this since the format got in the way more than helping us stay on track, so we might have been better off using a more visible and more organized task tracker.

Kyle – It was hard for me to play around with the hardware due to the compatibility of my computer. I could've prepared an adapter so I can tinker around the hardware while with the team instead of having to bring everything back to my place which prevents others from using it. I also would have managed my time better when it comes to paper work because sometimes I do forget (not often) about it and have to rush it in meetings.

Charles – It was hard to join this project in the middle, and I don't think I did enough to thoroughly integrate into the project's progression. Next time I would communicate more with the client to ensure that we can leverage prior work in a way that will elevate our project instead of being constrained to the work that has already been accomplished.

**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!