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: ___Shining Sky_____

Team members: ___Rosze Voronin, Ashleea Holloway, Skyler Hanson, Logan O'Donnell____

Course number and name: _____CS486C_____

Semester: ___Spring 2022___ Date this reflection completed: ____5/2/2022_____

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.
We used an informal process; basically we divided the development into certain modules and people worked on the modules they chose to take. However we eventually evolved to have our head programmer working on nearly all of the code while one person worked on aesthetic changes and the other two worked on other deliverables.

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?
In the end the process did work and we finished our project. However we think that we are dissatisfied with the way it went because there were a lot of difficulties. Some modules (database) took too long because those that worked on them couldn't get them working properly, and in the end the burden of writing code was not distributed evenly like we would have wanted.

What changes might you make in your development process if you have it to do again? More structure? Less? Different process model?
We think that we would use more structure, or at the very least trading off modules when problems are found. Also, we would have liked to experiment with our toolchain more and become more familiar with what we planned to use so that we could have considered reevaluating some of the more troublesome frameworks (like React Native itself) or just worked on learning them a bit more before having to perform.

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.
  - Atom - as an IDE for writing code
  - Visual Studio Code - as an IDE for writing code
  - IntelliJ Idea - as an IDE for writing code

- Version control: How did you manage your codebase?
  - GitHub - we used GitHub and GitHub branches to control merging/versioning of all code

- Bug tracking: How did you keep track of bugs, who was working on them, and their status
  - We kept an informal bug tracking system. Largely any bugs were just written down in our communications channel (Discord) and we tried to fix them as fast as possible to avoid any getting lost.

- UML modelers and other miscellaneous tools:
  - Microsoft Office - used Office shape/line tools to assemble diagrams for presentations and some documents
  - LucidChart - used this website to create diagrams for a few documents, but not as much as Office tools

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 largely maintained a stance of "whatever works in the final product" when utilizing tools; we did not see a need to necessarily use all the same things to produce what we did, but it might have made things a little more consistent. We feel like none of our tools were overkill or superfluous, and we also couldn't think of any new ones we would have wanted to add. In the end the only major change would have probably to pick one IDE, since we largely used the same or equivalent tools between us all.

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 started out with an "everyone does everything as needed" model and gradually moved into an "everyone does everything as needed but certain people are assigned to certain things first" model. Essentially, when we divided tasks, it would be on whoever volunteered for them, and certain people began to consistently volunteer for tasks of a certain type based on ability.

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

- Regular team meetings? If so, how often?
  We had a weekly meeting every Monday and our weekly mentor meeting let us briefly chat about anything else that came up as well.

- Impromptu team meetings? If so, roughly what percent of total team meetings were of this sort?
  We had a few of these, but they were not common. Impromptu communications were taken care of over Discord and were largely kept to text conversations. 10% or less of meetings were impromptu.

- Emails to all members? If so, explain briefly: about how often, what used for?
  We never used emails to all members except for in cases where Google Docs were being shared. The equivalent function was to use an @everyone message in Discord to ping all the other members of the chat. We used these pretty often, maybe once or twice a week, and it was largely to inform everyone of anything that everyone needed to know. Examples are documents being finished, issues that we need to work together to resolve, emails from the client, announcements, and reminders about meetings/tasks.

- 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?
  None of the software tools mentioned above were used to communicate or organize tasks. Our Discord server was our primary communication method.

- Other communication channels used? Facebook, wiki, text messages, phone conferences, etc.
  As mentioned, we used Discord. We made our own server with various text channels, and we also utilized the voice chat and screensharing features. We occasionally used text messages as a secondary method, and did some Zoom meetings with the client as well.

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 intra-team communication went fine. We had a few breakdowns, however we worked with our mentor to resolve them and avoid major consequences. People were largely good about clarifying due dates and keeping on top of things; most of the issues were interpersonal. The few problems we had were largely just a result of frustration and other breakdowns of communication.

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.
I think the only thing that could have solved our problems was to communicate a little more. Being proactive about solving problems and interpersonal conflicts would have kept our issues from escalating for sure. One idea we could think of is to have more space in meetings or a special type of meeting for check-ins and airing grievances, but time was tight as it is so we are not sure how practical it would have been, despite being a nice idea.

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!