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: Caribou Cams .

Team members: Samantha Muellner, Dongyang Yu, Shuyue Quai, and Keenan Swanson .

Course number and name: CS486 Senior Capstone Design________

Semester: Spring 2020 Date this reflection completed: ________________________

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 followed an Agile like process in order to develop our software: Plan, Design, Build, Test, Review, and Launch. We spent the first semester (fall semester) planning and designing our solution to our clients problem. The next semester we spent building and testing our solution, going back to the plan and design stages as needed in order to fix certain aspects that didn't pan out. At the end of the semester we have been reviewing our model and are about to launch it to the client.

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?

This process worked fairly well for all of us. It allowed us to fully design our model before hand so that, when we broke into separate groups and went to start coding it, we all had a good idea of what we needed to code instead of going in blindly

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

The model process our team followed worked well enough for us that we probably wouldn't change much, except perhaps to be a little more rigorous in the planning part.

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.

Eclipse and Visual Studios Code were used to code the different platforms

- Version control: How did you manage your codebase?

Uploading our code to github was used to help with version control, as well as manually entering the current version at the top of the code in the header comment

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

We didn't use a specific thing for bug tracking, instead we just worked together to solve them by hand

- UML modelers and other miscellaneous tools:

Draw.io was used to draw our UML models

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?

The coding process went well. The communication between teams was a little spotty, so this could have been improved.

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"?

Our roles were designated up front. Samantha Muellner was designated as Team Lead, Keenan Swanson as Customer Communicator, Dongyang Yu as Recorder, and Shuyue Qaio as Architect. Some of these roles other did, such as Samantha Muellner recording meetings and Dongyang Yu contacting our client occasionally, but mostly we stuck to our roles.

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

- Regular team meetings? If so, how often?

Yes, we had team meetings nearly every week.

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

Rare. We only had to have one, and it was over Skype. Otherwise we just used discord to discuss any issues.

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

None.

- 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?

Discord was our main software tool used to organize tasks and communicate.

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

N/A

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.

This went relatively well. Only a few times did people not respond to notifications about tasks being due or meetings being had, but usually everyone responded within a reasonable amount of time and nearly everything always got done in the end. Overall, the communication was good. We could all usually rely on each other to respond within the day, if not within a few hours, to a message.

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.

Overall, everything went fairly well. COVID-19 made it a little hard to have meetings between our group but overall everything went fairly well.

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!