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 instructor or team sponsor.

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: ___Skyward_____

Team members: __Gage Cottrell, Alexander Sears, Justin Kincaid, Chris French_____

Course number and name: ____CS 486 - Requirements Engineering II - Capstone_____

Semester: _Fall 2017__ Date this reflection completed: _5/3/2017_

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 development process. During the first semester we met once every two weeks as a team, then during the second semester we met once or twice every week. After a few weeks into the second semester we created "weekly sprint goals" that we would set out each week to complete. At the end of the "sprints" we would discuss our progress and what challenges remain.

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 were satisfied with our process, and it seemed adequate since we were able to complete our project with time to spare. If we could do it again we would have started setting “sprint goals” sooner, and met once per week instead of bi-weekly during the first semester.

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

Our development process worked well but if we had to do it again we would have added more structure. We went in with the mentality that people would do what they wanted or what was interesting to them, but perhaps a more strict assignment of roles and work would have made things flow a little smoother. In the end all of the work was completed, but with a little more structure we may have been able to complete the project even sooner.

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.

Our team mainly used Atom text editor for development. Atom works extremely well with web development since it is so lightweight and responsive - so we were generally happy with our choice. A few members used the pyCharm IDE for django development, which also worked well but is quite heavy and sometimes slow to run.

· Version control: How did you manage your codebase?

Our team used a private git repository on bitbucket to manage our codebase. We created feature-branches for newly created features, then had code reviews to decide if features were ready to be pushed to our master branch. Looking back it seems that we may have been able to speed up development by just using 1 or 2 branches for our version control instead of a new branch for each feature. This is mainly because our project size was relatively small, and getting complicated with git branching is more work than needed in these cases.

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

Because our team was consistently making changes and testing code, anytime a bug was discovered we would immediately work as a team to figure out the cause and fix the issue. We did not have very many bugs because our tools and frameworks were rock solid, but the ones we did have were quickly fixed by any member of the team. Usually, we would figure out who created the bug by looking through our repository's commit history and then having the bug creator fix their own mistake.

· UML modelers and other miscellaneous tools:

For diagrams we used Draw.io and Lucidchart because they are free and integrate well with google drive. We also had a member of our team that was experienced with photo editing in PhotoShop, so we were able to create some really good looking graphics using premium software.

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 main issue with coding that we had was utilizing git in the best way possible. Some members of our team had limited experience with version control, which lead us to work a little slower than ideal. We also created many branches in our repository which lead to some confusion as well. If we had to do it again we would probably spend a little more time getting the entire team up to speed with using git and version control best practices.

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 had official roles but in reality our team worked with an attitude of “do what you are interested in doing”. Our team was a group of calm and understanding members, so all of the work was completed with members doing what they felt like they could do. When deadlines approached we stayed in close contact and let the team work out who would do what. For some assignments we assigned specific parts to certain members, but for others we just went with the flow.

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

- Regular team meetings? If so, how often?

During the first semester we had team meetings every other week. Because the first semester was quite slow, we felt that bi-monthly meetings were most appropriate. Looking back we probably should have done weekly meetings in the first semester. During the second semester we met weekly, and sometimes two times per week. This seemed like a good amount of meeting times, but was harder to organize and pull off every single week due to scheduling conflicts.

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

Our team rarely had impromptu meetings since we had planned to meet on certain days in advance. There were a few days we would meet the day before the design reviews to get a little extra practice with presenting and communicating as a team - but these were quite rare.

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

We communicated with our sponsor / mentor with emails that were forwarded to all members of the team. For the most part we did not use email for team communication. Our team leader would cc the entire team on critical emails between sponsors and mentors.

- 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?

For discussion within documents we would use the google document chat, which worked well. We tried our hand at using the task management tool Trello, but it seemed a little bit like overkill for the size of our team and project.

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

For communication we began by using Slack, but we transitioned to the chat application Discord in favor of the layout and features Discord offered. We would rarely use other communication methods such as text-messages, but they did happen. We tried to keep all

of our communication in Discord so that everyone would be in the loop and on the same page.

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.

In the first semester we had a 5th member of our team that felt like they were "excluded" from our group. Unfortunately this 5th member never discussed anything with our team or faculty-mentor before being transferred to a different team (which they also left). If we could go back in time we might try to figure out exactly why this member felt "excluded" from our team and resolve that issue. However, our team seemed to function the exact same - or better - without a 5th member. The simple fact of the matter is that our project did not warrant 5 different developers. Once we had our 5th member leave our team, our team seemed to overcome all other issues and grow to an even stronger and more determined team.

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.

If we had more structure when it came down to assigning work/tasks to members things would have gone slightly better. If the team leader was assigning specific tasks to specific members based on each member's skillset, assignments may have been completed in a quicker and more efficient process. Again, it would have been beneficial for us to have had weekly meetings in the first semester instead of the bi-weekly meetings we planned.

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!