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: Git-OSS-um

Team members: Stephen White, Gary Baker, Van Steinbrenner

Course number and name: CS 486C: Capstone Experience

Semester: Spring 2019
Date this reflection completed: 4/24/19

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.

- The team decided to go with one week sprints in order to get our project done. Every week we would meet three times: once for our client meeting/mentor meeting, our formal team meeting, and our technical meeting. When we met, we would discuss what each of us has done, what worked, what failed, and what still needed to be accomplished. Through this iterative process, every week we would have something new and improved

for our client, making sure he was an active participant in the development of the web application.

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?

- The development process worked very well. Our sponsor as well as our team were very satisfied with this development process because there was consistent progress and always a sense of accomplishment as well as new features to show the client at weekly meetings. This process worked well for this project because it kept the teams progression on track and since our client has a highly technical background, it kept them happy throughout by being able to see new pieces of the product.

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

- We would pretty much keep the development process the same if we had to do it again. Our process kept both the team and the client happy throughout, which kept the entire process mostly smooth.

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 source code was created through Visual Studio Code due to its ability to easily highlight django syntax, and quickly create folders/documents. It was lighter weight than atom and completely free to use.

· Version control: How did you manage your codebase?

- We are using Git as our main source for version control. The source code is in a repository on GitHub, where we push our changes and keep track of our codebase. Each member of the team had a separate branch to develop on and push changes to. When we were satisfied with our respective branches, we would sit together and merge the code into the current working branch.

.

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

- We started by making use of the GitHub issue tracker to accomplish these tasks, but as time went on we made use of our slack and discord channels to discuss what needed to be done. Ultimately, we ended the semester with a few things left to do on the project, listing these tasks out on a whiteboard, and tracking their progress on slack. We made major efforts to keep the entire team in the loop when things needed urgent attention, and made room on Thursdays in our technical meeting for the collaborative development of the software so nobody fell behind on the bugs that needed to be fixed.

· UML modelers and other miscellaneous tools:

- We have used the draw.io platform to design our diagrams and models. It made it simple to throw together simple as well as complex diagrams for our 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?

- There were some issues with getting everyones machines set up for development. It seems that we could have come up with some better solutions to this problem, but in the heat of producing the product, we focused on getting the code put together on the machines that we could instead of focusing on the machine that didn't work. Something we could have tried was seeing if there were any opportunities to rent a laptop from the university as a quick way to get around this issue. We think that the tools we used were all necessary and that they did not create any avoidable challenges.

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 team has roles assigned to each team member. Stephen is the release manager, semester one team lead, customer communicator, and back-end specialist. Gary is the front-end specialist and semester two team lead. Van is the main editor and webmaster for the project. Documentation is split up evenly between all members of the team.

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

· Regular team meetings? If so, how often?

- We had team meetings twice a week usually, one meeting with our mentor, and usually one meeting a week with our sponsor. All of our meetings are regularly scheduled.

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

- There have been no impromptu meetings. Most of our meetings have been regularly scheduled. In semester two, the capstone session meets irregularly, so we would spend the allotted time as an additional meeting to work on the product or documentation.

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

- Emails have been used to communicate more with our mentor and sponsor. We would email our mentor and sponsor for further clarifications upon our product or documentation. The frequency of our emails would increase when we would have questions regarding documentation or the functionality of our product.

· 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?

- We did not use the GitHub issue tracker very much to communicate organized tasks, considering we took plenty of time out of our week to communicate what needed to get done in person. Our main platforms to keep track of what needed to be done were Slack and Discord.

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

- Slack, Discord, and Text Messages.

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.

- Intra-team communication was great. There were no communication breakdowns and all members always responded within a reasonable amount of time.

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.

- The main thing we would do to improve is to create a more formal task assignment system that way assignments would be in a central location and easy to find in case a member forgets what they were to do.

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!