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: Team Strata

Team members: Sean Baquiro, Matthew Enright, Jorge Felix, Tsosie Schneider

Course number and name: CS486c Capstone Experience

Semester: Spring 2016 **Date this reflection completed:** 5/5/2016

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 used an Agile development process, particularly SCRUM, because it provided us with a regular rhythm, provided transparency in tasks and progress, and kept us honest about deadlines. Due to the changing nature of our client's requirements, the SCRUM methodology allowed us to quickly adapt to the changes.

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 enjoyed using the Agile/SCRUM development process and it worked very well for this project. It allowed us to work in a flexible, iterative, and incremental process to achieve our goal and encouraged collaboration while working on tasks. We especially liked using Waffle.io, a SCRUM based project management tool that allowed us to connect directly to our GitHub project data and carry tasks from creation in a product backlog through to completion.

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

If we had to do it again we would probably try to align our development process even more with the SCRUM methodology. During the semester we met on a weekly basis to discuss with our mentor any problems we were facing, what we had accomplished that week, and what we were planning on working on the next week. To align more with the SCRUM methodology in the future we would have had quick fifteen minute meetings like this between team members on a daily basis, which would help identify problems better early on.

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.

We used the Microsoft Visual Studio IDE for C++ development with its built-in language support and the PyCharm IDE for Python development for its code analysis and unit testing support.

- Version control: How did you manage your codebase?

We used GitHub for version control as well as task management and documentation.

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

We used Waffle.io to track bugs and issues with development. Using Waffle.io any team member could create an issue and the rest of the team could track the issue and make updates from creation to resolution.

- UML modelers and other miscellaneous tools:

We used the Lucidchart online tool to easily create UML and flow chart diagrams. Google Drive is also a miscellaneous tool that should be mentioned, as we used it to store and collaborate on all of the deliverables throughout the semester.

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?

GitHub is a relatively easy and effective way to contribute work and track changes and we didn't really experience any issues using the repository for version control. It also allowed us to easily transfer our work to our sponsor after completion which was a nice benefit. Waffle.io was also a great development management tool that allowed us to create and track tasks and issues throughout the semester and provide our progress to our mentor at any time. Using Google Drive made it simple for all team members to work on deliverables and make changes at the same time, and along with Facebook Messenger it made it so that we were able to fluidly communicate and work on deliverables with each other without actually having to meet in person. There are certainly alternatives to these tools out there that we could try next time, but we didn't experience any real problems while using these for our project.

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 a clear distribution of team roles from the beginning, assigning the positions of team leader, release manager, recorder, and software architect to each member upfront. Along with these roles each team member also had the role of developer on this project.

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

- Regular team meetings? If so, how often?

We had regular weekly team meetings with our mentor to discuss any issues, our progress, and planned work. We also planned a handful of team meetings with our sponsor, Ryan Anderson, at USGS Flagstaff.

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

We had impromptu team meetings when necessary to work on specific issues and practice dry-run presentations for design reviews and UGRADS. Roughly 25% of team meetings were impromptu.

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

Our team really only emailed when sending deliverables and receiving feedback from our mentor and sponsor.

- 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 used Waffle.io to create and track all tasks and issues. Any tasks identified in team meetings were immediately created in Waffle.io.

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

Our team used Facebook Messenger as our main method of non-team meeting communication to discuss anything related to the project.

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.

Our team didn't really experience any breakdowns like those mentioned. By using Waffle.io we also created tasks upfront for all of our deliverables throughout the semester so everyone in the team always knew when the due dates were. By using Facebook Messenger we were also in constant communication with one another and able to discuss and resolve issues quickly.

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.

We didn't really experience any breakdowns in communication, as mentioned in the answer to the previous question. That being said, communication could be improved and the risk of any breakdowns like those mentioned occurring could be lowered even more by aligning our development process more closely to the SCRUM methodology. Having daily fifteen minute meetings with the team to discuss any issues, our progress, and next tasks would definitely reduce the chances of any breakdowns occurring.

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 bumpy ride…usually it's a combination of factors. And always, regardless of project or team, there are things we could have done differently to make it flow better. Recognizing those things through thoughtful reflection post-facto is the key to improvement!