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: Team Gaming Ed

Team members: Brett Lewerke, Grace Hsieh, Chayson Spigarelli

Course number and name: CS486c Capstone Experience

Semester: Fall 2022 Date this reflection completed: 12/14/2022

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.

Our team structured out the development process through Agile methods. We would code in weekly sprints approved by the majority of the team.

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?

I think that this process went very smoothly. However, there were some weekly sprints that we tried to undertake and they took longer than a week to complete. I think having more foresight and knowledge of the code we were making would have made this process easier.

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

I think that our process was very good. It would have worked better had we had more knowledge of the technologies we were using. I do not think this would be an issue in a workplace setting because we would already have knowledge on the technologies we were using.

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.
  - Visual Studio
  - Visual Studio Code
  - Unity Editor
  - .NET Core
  - XAMPP
  - MySQL
  - AWS

- Version control: How did you manage your codebase?

We used GitHub for our version control. We would always have two up to date branches that would match each other. Any new update we made to our codebase would be done in a temporary branch. For Unity, it had issues with updating to GitHub and we kept 4 local backups at one time for version control.

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

  Our bug tracking method was a little rudimentary and we would track them in our Discord chat. We would delete comments when the bugs were fixed.

- UML modelers and other miscellaneous tools:

We did not use the obsolete method of UML diagrams as they can change too much. We used very detailed task reports of the tasks that we had each week. Our group coded very specific portions not very related to each other so running into naming conventions was not an issue. Brett also used a large whiteboard in his room to keep track of things to be done in Unity.

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?

I think that not doing UML diagrams, but some sort of method tracker would have been a good idea. I think sometimes our group would forget when we needed to do something because there was so much to do. Our Ghantt chart was a little overkill because it's hard to predict each week or phase to do from so far out in the future. I think that using some sort of diagram on how the scope our of project worked would have helped some group members stay on track and code for the future.

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"?

Grace Hsieh was our team leader and she was in charge of sending all the documents, hosting team meetings, making the skeleton for our documents, keeping in touch with the sponsor, and so much more.
Chase Spigarelli was in charge of making the template for all of our task reports and styling our team website.
Brett Lewerke was in charge of architecting our whole project and teaching some group members how the technologies worked.

Everyone coded and added to the task reports, documents, and other things.

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 meetings on Tuesdays at 2:30 on discord.

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

We had no impromptu team meetings. Grace and Brett were in other class groups together so they would sometimes talk about new things that needed to be done and then pass it along to Chayson.

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

We did not really send emails. But we did talk ALL the time on discord, almost every single day. They were rarely times when at least one chat did not go through a day.

- 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 for strictly speaking to one another, we mostly kept on discord.

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

Yes, our main form of communication was through discord. In comparison to the type of casual talk that we spoke it was almost like Microsoft Teams.

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.

We had breakdowns ALL the time. Our team suffered an issue in the beginning of the semester where we almost had a team member removed from our group. However, we worked past it and honestly after the capstone conference everyone was happy as a daisy. We did not really forget anything that was due except a few task reports, however our team leader Grace is awesome and she would do it for us no questions asked.

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 that a better structured team knowledge would have been better for the project. Our team sometimes argued over the functionality of our project and some of the weekly tasks. I think that we should have stuck with the majority rule of doing things in our code but sometimes the University forces you to do things that's out of your control. I think to avoid breakdowns we needed a set requirement for the project and an understand of what we were about to accomplish.

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!