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: F1/10 Yellowtails

Team members: Bowen Boyd, Hanyue Wang, Kyle Watson, Jordan Wright

Course number and name: CS 486C - Capstone Experience

Semester: Spring 2020 **Date this reflection completed:** May 5th, 2020

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.

- We choose to go with an Agile approach for this project. We decided to use Agile to meet the needs of the project when it changes. This ensures that we are able to meet the users needs.

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?

- Tasks were handled dynamically in a sometimes ad hoc fashion. More structure in handling task assignment would have been useful.

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

- More clearly outline requirements (early on) in order to more effectively assign tasks.

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.
  - Vim
  - Command line
  - Visual Studio Code
  - Geddit
- Version control: How did you manage your codebase?
  - Github repository
- Bug tracking: How did you keep track of bugs, who was working on them, and their status
  - Cards on Trello board
    - each team member handled various bugs (first come first serve)
- UML modelers and other miscellaneous tools:
  - PyQt5
  - Draw.io
  - Online Google Services
    - Slides, Docs Folder
  - Trello

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?

- Version control was handled via Github but with only a master branch. Therefore, dev branching in Github would have been useful.

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"?

- Team lead was a clearly denoted role
- Other roles were distributed on an as-needed basis

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

- Regular team meetings? If so, how often?
    - Weekly team meetings
- Impromptu team meetings? If so, roughly what percent of total team meetings were of this sort?
    - 5 impromptu meetings (~14% of team meetings)
- Emails to all members? If so, explain briefly: about how often, what used for?
    - Email not used for team communication (Discord was used)
- 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 used for team communication
- Other communication channels used? Facebook, wiki, text messages, phone conferences, etc.
    - Text messaging and Zoom

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.

- Overall there was good intra-team communication
- Team meetings every week helped in continual communication and keeping track of team member progress

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.

- More strictly follow the Trello board tasks
- Potentially use Github project boards since the Github repo was closely followed by each team member (more so than Trello).

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!