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: SARCI

Team members: Dylan Woolley, Jabril Gray, Randy Duerinck, Vidal Martinez

Course number and name: CS 486: Senior Capstone Design

Semester: Spring 2022   Date this reflection completed: 5/3/2022

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 applied an Agile methodology with continuous development and testing. We believe that an efficient, well communicative team is important to delivering a clean project.

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?

Our team faced confusion and frustration at the start of the capstone, specifically in the process of handling completion of deliverables and code sections. Not planning well enough ahead for the delegations of tasks caused issues for the team. Our team did adjust and become more comfortable in our process cycle.

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

Our team, if we could do it again, would plan ahead more and manage our time better. We would include more structure and use better communication, but still follow Agile. Specifically, better communication with detailed task lists and improved organization strategies. Above all staying true to the standards set by the team and amending the standards document as needed. Additionally, the standards document should be structured better with higher detail to force teams to define strong standards.

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 2019
    - IIS Express
  - MySQL database system
    - MySQL workbench
  - Amazon Web Service Helper Deployment

- Version control: How did you manage your codebase?
  - GitHub
  - Discord

- Bug tracking: How did you keep track of bugs, who was working on them, and their status
  - GitHub
    - Our team communicated bugs discovered over Discord

- UML modelers and other miscellaneous tools:
  - UML, Class, and other Diagrams
    - Google Drive - Draw.io

        - Diagrams.net
    - Canva.com for gantt charts

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 primary issues with coding and organizing code were planning ahead at the start of the capstone project as well as improper use of gitignore in the GitHub repository. If our team had spent time studying and practicing with Visual Studio and C#, then we could have had a better experience. Gitignore was meant to exclude files not necessary to the code base. Using Amazon Web Service to host our website and database also brought on issues. Due to the regulations set by our client, certain permissions prevented our team from completing and testing what was necessary for the project causing delays. The only way to avoid this issue would have been to communicate with our client sooner to handle issues with permissions.

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 gave each of the team members roles
Leader- Dylan Woodley
Recorder - Randy Duerinck
Version Manager - Jabril Gray
Vidal Martinez - Architecture

We end up switching to everyone does everything as needed, only keeping the roles of Recorder and Leader

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

- Regular team meetings? If so, how often?

Our team met two days a week for anywhere from one to 3 hours each session. At the start of the project our team did not implement this, but we found that more regular meetings helped in productivity.

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

Impromptu meetings made up approximately 20% of the total meetings. These were helpful for wrapping up deliverables and addressing issues our team faced.

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

Our team leader handled all primary communication and included all members as a carbon copy. Emails were used for much of the communication between our team and the client.

- Software tools? Were any of the software tools you mentioned above (e.g. bug/issue tracking) used to communicate and organize tasks, e.g., in lieu of emails or other discussion?

Discord was the primary tool used for communicating between team members on bugs and issues. Zoom was used in addition to email for communication with our client on progress and issues.

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

Our team did not include any other communication channels outside of email, Discord, or 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.

Communication overall has gone well. At the start, as mentioned, there were issues getting into the groove, but overtime our team became more cohesive and efficient. There were some situations where team members forgot about deadlines or completing their task. Much of the problems in this context came from the struggle of balancing other classes, work, and capstone. The consequences only went as far as meeting with the mentor and all team members to discuss the issue.

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.

With the knowledge our team has now, we know more structure and communication would have made our process easier. We used useful tools and the general approach to communication was successful, however the frequency and level of detail our team followed needed improvement.

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!