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: Anubis

Team members: Frankie Berry, Steven Gruenewald, Marjorie Hahn, Riley Shelton, Matthew Siewierski

Course number and name: CS486C Capstone Experience

Semester: Spring 2017 Date this reflection completed: May 10, 2017

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 software development process was fairly random. It was more or less one of us claiming a task and then completing it. We did use a Trello in order to organize those tasks, but for the most part assignment was random.

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?

It could have gone better; the random process was stressful at times and we could have been more organized in order to get things done.

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

If we could do it again, it would have worked better to choose a more formal development process such as Agile. That way we could have been more on task with regular deliverables for the project.

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.

  DNN built-in development environment
  Atom text editor

- Version control: How did you manage your codebase?

  Github

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

  Trello for task management

- UML modelers and other miscellaneous tools:

  Photoshop for image editing
  Adobe Illustrator for logo design
  Lucidchart for creating diagrams
  Google Drive for sharing documents
  Slack for communicating

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?

Git integrations with DNN would have been helpful. DNN in general was finicky to use with all of the modules we were attempting to build. We would have gone with a different CMS if we had known DNN's issues at the start.

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 loosely defined roles, but for the most part everyone volunteered and did things as needed.

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

- Regular team meetings? If so, how often?

  Yes – weekly meetings.

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

  Not as often, most all of our meetings were planned. Maybe 5% of meeting were impromptu.

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

  Mainly to forward information from sponsors and team mentor as needed.

- 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?

  Trello for task management
  Slack for communication

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

Occasional text messages if we needed to get in touch with someone asap.

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 communication was pretty good, very few breakdowns.

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 could have benefited from a more formal task assignment system to keep everyone on track and busy.

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!