NAU–CS Team Project Self-Reflection Worksheet

Team Name: _The Disease Outbreaks Team_____

Team members: Abdulaziz Alhawas, Jean-Paul Labadie, Jordan Marshall, Luis Valenzuela

Course number and name: CS 486C Capstone Experience_____

Semester: ___Spring 2016_____ Date this reflection completed: 5/4/16

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 closest formal structure to our process was a mix between SCRUM and Prototyping. We had weekly meetings in which we would discuss what needed to be completed for a given week. Once tasks were determined each member chose a task to complete for the week. This was repeated for the duration of the project. We had an early prototype that we showed Tgen and refined every week based on their feedback. This is where some of those weekly tasks were defined. We never explicitly defined a development method it just naturally happened this way given the nature of the project and our schedules.

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?

This process worked well for the most part. Every weekly meeting gave us something to work on but if we had issues that we couldn't solve it was a slow process for getting help either through email or Slack. There was also instances of code conflicts early on in the project. Since we worked on our tasks independently we didn't have a way to check whether our code conflicted until we met.

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

As mentioned before the process worked well but we faced issues due to the frequency of the meetings. If we were to do this again it would be a good idea to have more than one weekly

meeting and maybe even have days where we all worked on the project together. This would help resolve issues that may arise like code conflicts and problem solving.

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.
    - IntelliJ was used by most of the team members for editing and writing code. It had the best integration with the tools that we were using to create the interface.
    - Netbeans was used by one of the team members. He was having issues with IntelliJ and JavaFx and found that Netbeans worked better for him.

- Version control: How did you manage your codebase?
    - Bitbucket was our choice for code management. We chose Bitbucket over Github and other alternatives because Bitbucket allowed us to create private repositories.

- UML modelers and other miscellaneous tools:
    - Gantt Project was used to create our schedules early in the project. Later we switched to using Creately to create custom schedules.
    - Creately was also used to create the diagrams for the project. Diagrams such as the architecture and use case diagrams were produced using Creately.
    - Slack was used for the majority of the communication within the team.

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?

In the early stages of the project we had issues with the repository due to the gitignore file. There were some files that we didn't include in the gitignore and we had a few issues arise due to that. Next time we would make sure that we had all of the proper files in the gitignore to avoid this issue. The tools themselves were never a hindrance but instead improved the way in which we accomplish certain tasks like code base management and team communication.

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"?

Early on the team roles that were defined for each of our group members. The roles were followed at the start but as the semester progressed we all just did what needed to be done.

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

- Regular team meetings? If so, how often?
    - We had weekly meetings where we discussed tasks that need to be accomplished.

- Impromptu team meetings? If so, roughly what percent of total team meetings were of this sort?
    - Impromptu meeting happened the most when we had major assignments due as needed. For the most part the meetings took place weekly.

- Emails to all members? If so, explain briefly: about how often, what used for?
    - Group emails were used when communicating with our sponsor or our mentor. This helped keep the group up to date with everything that was being discussed.

- 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 prefered using Slack to communicate over group emails. Slack was more immediate and just overall a more efficient method of communicating with everyone in the team at once.

- Other communication channels used? Facebook, wiki, text messages, phone conferences, etc.
    - Text messages and phone calls were also used occasionally but again, the best form of communication that we found was through Slack

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.

There were times throughout the length of the project that we would realize that an assignment was due for the course. We tended to be more focused on actually programming than on the written assignments. Meetings were canceled occasionally without the team knowing but this was just a lack of communicating the situation to the entire team. Overall we all knew what was going on throughout the project.

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.

The best way to improve the situations that we encountered would simply be to communicate with the entire team about an unscheduled event. The assignment due dates could also be easily solved by having someone tasked to keep track of what was due when.

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!