CS349 Written Comments

Jacob von der Lippe

December 17, 2017

1 Topic: Process

Our group decided to follow an Agile process model that allowed us to be dynamic in our methods for developing, testing, and distributing our mobile application. The Agile software model was ideal because it put an emphasis on customer satisfaction and early incremental delivery of our software. This meant that we could create small, highly motivated portions of the project in an informal manner. This made organization within the group much more convenient, as well as we were able to catch and correct inevitable bugs in a more timely fashion. From our book we learned that our highest priority was to satisfy the customer through early and continuous delivery of valuable software. I believe my group took this to heart from the beginning although we ran into some timing problems towards the middle of the semester with some of the code not being implemented according to our milestones.

However, we used this as an opportunity to embrace the second most important part of the process, which our book defines as: "welcome changing requirements, even late in development." A major turning point for our group was deciding to shift away from a multiple-platform application and instead focus our time and energy on only a single platform (Apple IOS.) This change may have shortened the overall distribution of our application, but in the end it strengthened the quality, lowered the cost, and increased the security of our application as a whole. We also made sure to increase the number of team meetings during the week and engage more often in face-to-face conversation about the project. This greatly increased productivity and we were able to finish our application on time with a few extra features that added to the stability and user-friendliness of the mobile application.

If there is one change I could have made to the project process, it would be to establish a better plan for risk management. Our group had run into a few problems with members not completing work on time or being unable to find resources that we had initially planned to have access to. In these cases I believe we did an okay job of working with each other and correcting parts of the project as needed to fit with our different schedules and resources. Although, it would have been a lot more efficient to have had a better plan for these incidents ahead of time. In our initial requirements document we would have provided more information about team member discipline on deadlines as well as alternatives to exterior resources (such as a database supplied by the school) if it turned out we would not have

access to them. Other than that, I am happy with the outcome of our project, and I believe we followed the idea of the Agile process model to the best of our abilities.

2 SE Code of Ethics: Principle 7

Definition: Principle 7 COLLEAGUES Software engineers shall be fair to and supportive of their colleagues. In particular, software engineers shall, as appropriate:

7.01. Encourage colleagues to adhere to this Code.

7.02. Assist colleagues in professional development.

7.03. Credit fully the work of others and refrain from taking undue credit.

7.04. Review the work of others in an objective, candid, and properly-documented way.

7.05. Give a fair hearing to the opinions, concerns, or complaints of a colleague.

7.06. Assist colleagues in being fully aware of current standard work practices including policies and procedures for protecting passwords, files and other confidential information, and security measures in general.

7.07. Not unfairly intervene in the career of any colleague; however, concern for the employer, the client or public interest may compel software engineers, in good faith, to question the competence of a colleague.

7.08. In situations outside of their own areas of competence, call upon the opinions of other professionals who have competence in that area.

My team and I implemented this topic throughout our project in order to maintain a strong and supportive working environment for creating software. As discussed above, we did struggle for the first half of the semester with coordinating times and meeting deadlines. Although, as we shifted back towards a more diligent mindset, we took on much more of the characteristics of Principle 7. We began to encourage each other to follow the directions established by our requirements document through coding tests (I assigned the group a test to develop a working UI in Swift given a week deadline. This is actually how we established the UI used in our final product.) These tests pushed us to advance our understanding of the language we were working with and inevitably gave us the push we needed to complete the project. We then established teams within the group to work on similar parts of the software. This way we could listen to and support each other's work. Similarly we increased the number of meeting times we had during the week allowing us to give credit to individuals and small groups for good work.

During the testing phase we were able to critique each other's work and give tips or ideas on how to improve each other's abilities for the future. This greatly improved the dynamic of the group as well as the software itself. Whenever problems arose, we were comfortable speaking with each other to find solutions. Even if those solutions meant getting help outside of our team (Dr. Bressoud.) Although we had a rough start to the project, I think towards the end we completely embodied the ideal of Principle 7 and created a safe and hard-working environment for all members of our software engineering team.