Final Report Part 2

Wes Lanning   
COMP 495 V1WW

Professor Moreland  
December 14, 2013

*How did your team learn to work together? What adjustments did you have to make along the way?*

Working together for my team came fairly easy, though it required getting to know how each of us worked the first couple of weeks to make things more efficient. I was lucky to have a teammate that was open to using new technologies and flexible with his communication hours and the tools we used. We both felt we communicated in the short term more effectively by using Google Talk/Hangouts combined with Team Viewer for collaboration. Using voice chat and screen sharing it made it easy to stay in contact and the actual process of learning to work together less of a burden than if we only had used Franklin Live and Unfuddle. Overall, it was almost as good as being there in person. Unfuddle was still an important tool, but more for long-term communication, such as weekly requirements and milestones. The other aspect that made communication easier (for better or worse) was only having two of us instead of three, as one member never showed up.

The biggest adjustment I had to make was defining Tony’s role in the project to being in a more specialized area instead of jumping around to various tasks. I realized a few weeks after the project started that he was being overwhelmed by the project and we talked about narrowing his focus to working on building the controllers, models and views for the project, while I handled the database and asynchronous communication areas. I think he still felt the project was still daunting after that, but it did help to alleviate some of the anxiety he was feeling.

*State what 3 project-related obstacles you overcame during the course.*

The first week of the course, we lost one team member, making our group of three, a group of two. That had its pros and cons. It meant we would be able to communicate easier and with more one-on-one assistance, but it also meant that we would have to take on more work that could be distributed to our missing third teammate. That meant we would have to eventually drop some of the features we wanted to implement as we started to run out of time. I think we coped well for missing a third member, but it would have been nice to have three instead of two.

Another obstacle we had to overcome was figuring out how all the various pieces of technology we were working with would fit together. I had used all of the pieces before in separate projects, but I had never used them all together or all in Python. However, I was confident, they would all work together from my experience with how they performed. After doing some initial prototyping and testing of the tools, I came to the conclusion that they would have no issues working together we solidified our project design.

The final obstacle I touched on already and that is integrating my team into working with technologies and tools they had never used before. Tony had a little experience with using JavaScript, Python and Git before, but nowhere near what he would learn on the project. However, he was eager to learn and jumped into the project without reluctance. When I started to see his anxiety about the project’s feasibility a few weeks later I talked with him about narrowing the scope of what he was working on so that he would not be overwhelmed and assured him we would have a working project by the final presentation. I know he says he was still lost at points over the project, but I think that was more regarding the entire project as a whole, not what he was working on himself. In large projects though in the real world, it is common for developers to work on a specific part and not be an expert in every area and I conveyed that to him. I know he wanted to learn as much as he could, but I can understand how there was just too much going on and not enough time.

*What did you learn from working on a software development team?*

Probably the biggest fundamental I learned from working on a team is a better trust and understanding of the people working with you. Probably 85% of the projects I do on my own are entirely my own code and without formal collaboration with others. The other 15% is rather spread out and not always consistent. As a developer, there is always an urge to want to do everything yourself, but sometimes you just have to let go and trust the team you work with to do the job correctly. That does not mean you assign them at task and forget it after that. It just means you give them something to do and then guide them through the process in order to complete it according to how you envisioned.

Overall, I would say soft skills are just as important as any technical skills. If one cannot communicate effectively with others, it is unlikely one will be able to hold a job in a collaborative development environment. In the age of large projects, the “lone coder” is a rarity and so developers must learn to communicate or perish. I would surmise that the soft skills are perhaps even more important to many companies than the technical skills because if one cannot communicate with others, then even a brilliant developer cannot save a project.