**Release 1 Team Reflection**

|  |  |
| --- | --- |
| **Team** | s361-02b, Team B  Jedd Haberstro, Obaseki Idemudia, Gregg Miller, Johanna Calderon, Jenny Zhen, Richard Sperrazza |

For the most part we abided to the team roles that were given to each of us. There were no features, deadlines, or documents that fell between the cracks which we believe is a good sign. We believe that things went relatively smoothly for our team for two primary reasons. First, all team members work together very well and we all like each other. We’ve all come to the team meetings and we’ve had no personality clashes that would hinder our ability to get work done. The second reason we were able to remain on target was that we thought carefully about our design and laid out the large architectural components ahead of time. This made integration extremely painless (there were almost no bugs when combining each component together which is almost unheard of), because it was clear from the design the responsibilities of each component.

In terms of workload, even though we remained relatively within our own roles when considering documents that needed to be submitted, etc., there was a lot of cross-role teamwork. This allowed each member to capitalize on their specific strengths, even if that strength wasn’t related to their assigned role.

Another key aspect of our success in release 1 was our ability in choosing the right features to include. We believe we struck the optimal balance that allowed us to deliver on release 1 without too much crunch without putting too much work off for the final release. There was the appropriate amount of workload, the workload was well-defined, and that allowed us to stay motivated by having well-defined goals.

We also found team meetings highly effective. We would often work together and this was beneficial to us because of our cross-role participation. It really encouraged contributions from everyone, and allowed us to utilize another members strengths by consulting him or her if we believed they could that person could help you better solve the problem you were working on.

Finally, there were some things we could have done better. The first was that there was still an occasional communication mishap. For example, at one point we had two member working on independent code for the same feature. This obviously was not ideal because we had to throw away one of their works, and so it was effectively wasted time. Another area we believe we can improve is our unit testing. For the most part, we put off writing unit tests until the end of the release cycle. We believe we can improve this by following a more test-driven development method and write unit tests side-by-side writing the actual implementation.