**Project Postmortem**

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

Looking back on the work we have completed, we have agreed that everything went fairly smoothly. There were a few rough patches, but for a randomly assigned group with nothing more in common than the class we were taking, we worked together quite well and completed the project satisfactorily.

We would not have planned our releases differently. We could have started work on the manager functionality at an earlier time (so that we could have realized a small, but crucial design flaw earlier in the development process), but we believe it was still appropriate to not have had that functionality until the final release.

Some risks that we encountered include team members unable to show up for the meetings and team members unable to complete a component of the project that the rest of the team needed to continue. Risks that never occurred include strictly tying work to roles of team members, e.g., testing being done by only the Test Coordinator. An unexpected problem occurred when the last component of the project, that had to be completed a few days before the final release, was not committed early enough. Thus, the errors in the code for the component were rushed to be fixed. In future projects, risks should not be accounted for lightly. Having tasks assigned to be started and completed early is better in any situation.

One of the most challenging aspects of the project was being able to meet self-set deadlines. Another challenging aspect was finding a job that was fitting for each team member, since all the team members had varying levels of experience with programming and dealing with large projects, in general. For some of the team members, this was the first large project to work in a more realistic team environment.

There were no changes to our process made after the initial release that had any effect. Since we often switched roles around and divided work as was needed, there was not a firm process in the first place.

Setting goals early and often helped us stay on track. When we had a large work load simultaneously, i.e., the list of features that had to be worked on for Release 2, we made certain to divide the work amongst members right away. Everyone could get started on their tasks early and find any issues that needed to be dealt with. Extensive planning helped avoid major integration issues.

For any future teams, our advice is to plan ahead, make goals for each week, commit important code early and often, ask teammates to review your code, don’t slack off in any week, assign parts so that everyone knows what to do, hold a meeting at least once per week where everyone must attend, make sure everyone understands the requirements for the entirety of the project and the assigned parts, and determine and complete the crucial components needed for the entire system. Don’t get too close to the deadlines to have the first release ready, so that more time could be allotted for the next release. Lastly, always aim for effective communication among team members.