



Imperium

Team 15

Jeremy Potratz
CJ Enright
Anthony Popilov
Peter Reckamp
Jacob Szarzynski

What went well

This sprint we were able to accomplish all of our user stories and tasks assigned to us, so we were able to be very productive throughout the sprint. We were also able to chip away at some of the things in our non functional user stories (like the design of our site), which will help to make an overall better end product. As a whole, our team was able to use git pretty consistently with only a few minor issues. Git proved invaluable when we wanted to reference deleted code that suddenly become relevant again.

Backend/frontend error handling

We were able to get errors communicated from the backend to the frontend in a pretty generic way that we hope will work for most situations. If there was an error, the JSON response sent to the front end will have a non-null error field with the error to show the user. So far this have proven to be easy to work with and flexible enough to work on all of our views.

Communication

We used a Facebook Messenger group chat to coordinate, and while that's a simple method of communication it ended up being very effective. We always knew when meetings were going to happen, and were always able to hold each other accountable.

What did not go well?

Part of the reason that we were able to finish all of our user stories this sprint could have been due to us not taking on as many user stories as the guidelines suggest. We didn't take as many this sprint because the ones did choose were large scale stories that future stories all rely on. For future sprints we will be closer to the suggested number of stories.

Git

On a few occasions we ran into some merge conflicts. Of course, these are to be expected and can be dealt with quickly, but some of those times we accidentally took the incorrect action and deleted the HEAD version instead of the outdated local version. While this was easy to fix, it took us a bit to find out what had happened and rectify the situation.

How should you improve?

1. Take on more user stories so that we can have a good number of features by the time sprint 3 comes to a close. Last time, we missed a point for not having enough user stories. While we did have more than enough hours to keep us occupied, this should be remedied. In the future, we must include more user stories to combat this issue.
2. Take care when dealing with git merge conflicts to avoid headaches of accidentally deleted code. Otherwise, we will waste time figuring out exactly what went wrong and which set of code to keep/delete. This is annoying and can be solved if we are just careful with pulling and pushing to GitHub, while also coordinating with fellow teammates to make sure everything goes swimmingly.
3. Use meetings more effectively to coordinate and set the pace for development. Last sprint, we procrastinated a little too much on finalizing our user stories and creating the testing document. As a result, our final meeting took much longer than it should have and we crammed too much work into one occasion. With more effective meetings, we can schedule work more efficiently and prevent a hefty last-minute workload.