Individual Reflection and Software Leadership

Mitchell Meier

December 11, 2020

When looking for ideas for our project, my group narrowed in on a couple of specific goals. The first goal was to make efficent use of our limited time. We knew that with only four months, we would have significantly less time to develop our application than our competition in the market we were trying to capture. So to counteract this, we decided to target a subset of our audience, in order to not only focus on implementing specialized functionalities that were less likely to have been done before, but also reduce the amount of functionalities total that we would have to implement. This is how we came up for the idea to tailor our project around S&T students

The other major goal we had also played a part in the direction of our application. The goal was to make the development and collaboration process as easy as possible. To do this, we settled on a platform and development environment before we began planning out the contents of the application. Before making this goal, we had considered splitting up and researching different development environments, to see if we wanted to take the time and resources to learn and use them for our main project. However, in the end we decided to just go with an environment that we were already familiar with, that we believed could be used to implement most of what we wanted

Both of these goals were major parts of our projects, and both goals were made out of ambiguous questions. Especially with the second goal, we didn't have all the information that could shape our decision when we made that goal. Just like what was mentioned in the webinar, we didn't have 100% of the information we needed to make the decision. But we made the decision anyway because trying to get 100% of the information we needed would be a less efficent use of our already limited time than getting started on acheiving that goal. And in addition, splitting up that work between multipule people would lead to an eventual state where most people's work would be discarded when the final platform was chosen

The idea of an ambiguous decision was something I learned a good amount about during the software development process. My thinking tends to be analytical, and when I don't have the full picture of a problem, its hard for me to make an educated decision. I learned quickly however that getting information can be like an exponential relation, where you learn much more when you first start researching, but as you dive deeper into research, the amount you are learning decreases as it takes you longer to find unique, relevant information. So at some point, it is more time efficent to switch off from research into implementation, and that usually comes long before you are 100% educated on the topic. My teammates were already good at making ambiguous decisions, and seeing how they approached problems like these helped me myself become better at analyzing these situations as well.