|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | **Challenge** | **How did we overcome?** | **Lessons Learned** |
| Domain | - Considering the fact that some of us were international students and the fact that education systems are so different in our home countries, understanding the domain was a challenge. | - Regular meetings with assigned project mentor helped us overcome this challenge. Also, the Data Model diagram gave us more insight into the domain. | - We believe the team should invest enough time to understand the domain. In the real world its not always as simple as this. Domains could range from Automobile and Manufacturing to Securities and Insurance. Devoting a considerable amount of time upfront will surely save the team a lot of late-nighters later on. |
| Requirement Gathering | - In many instances, the team would face difficulties in understanding the requirements. And each ones perception of the requirements would be different. For example, there were instances where the visualizations developer would use an API which is different from what the standardized report developer is expecting. And this causes delay in the deliverables. | - Although we did have challenges, with the right attitude and perseverance, we were able to have all our understanding of the requirements ironed out. | - It is crucial to iron out the requirements in the early stages of the project. In many instances, the clients would have doubts about their own requirements. So it is important to have regular meetings with the clients to understand what the expected outcome is. This will lead to a quality deliverable. |
| Programming Language | - Although some team members were adept at Ruby programming, some others in the team were novice at programming. And with little knowledge we started the project. | - I think everyone took a step back and tried to understand the programming language a little better. We understood the object oriented programming principles before diving in Ruby programming. | - It is during this step that we realized that we need to be strong with our basics. In this case, we needed to have strong understanding of the Model-View-Controller architecture. |
| Tools | - Some of us were using a client IDE to run the Ruby code and some others were using online IDEs like Cloud9. It took some time for each of us to figure out what's the best IDE for each of us. | - Trial and error, we realized what’s best for each of us. | - Upfront, during the inception phase of the project, we have to identify the development environments and how we are going to branch/merge the code. These standards need to be set and followed for effective communication in the team. |
| Effort Estimation and Effort Assignment | - At the beginning of the project, we did not take enough care to understand each of the team members strengths and weaknesses. This lead to re-assignment of duties mid-way into the project. | - Mid-way into the project, the team was facing difficulties in completing their assigned tasks. However, the team was cooperative enough to shuffle the work around. We did some reassignment and we all were happy about it. | - During Inception phase of the project, we have to ensure that team meetings cover the strengths and weaknesses of each of the team members. Without understanding the problem statement or the domain, many times we find ourselves owning tasks that we eventually find we are not comfortable with. |
| Deliverables | - There are many aspects to this project. One is the fact that we had to have the project up and running. At the same time, we had to intertwine the coursework to the project. Intertwining the coursework meant that there were a lot more deliverables (Project Documentation Folder) than just running the project. Many times, these final deliverables are available upfront and we could've just missed seeing them. | - Towards the end, we had to hush and rush our deliverables. In a short span of time, we had to think about the work we did in the last couple months and come up with a lot of documentation. | - It is important to meet with the client (or professor in this case) and the project mentor and discuss the final deliverables. Sometimes it is readily available and sometimes it’s not. It is important to drive the project from a deliverable perspective so that we achieve small chunks of deliverables as and when rather than delaying them till the end. |
| Team Dynamics | - Although we were meeting every week, we felt that we could improve on communication and keeping everyone else in loop. Sometimes, we would individually email the mentor and some other times we would group email. I guess we could've improved on the communication channel.  - Working in silos. Although there was enough dependency between the work that each of us did, we felt that we worked in silos. Only during the team meetings did we meet and discuss our deliverables. | - Some of us have a challenge in communicating in English owing to the fact that we are international students. However, the team was very supportive in understanding the background and we supported each other. | - In cases where we feel there is a lack in communication, one of the team members has to take the initiative to drive all communications. These projects need to be developed as a team and it becomes tough to achieve desired output when we work in silos. Hence the lesson we learned here is that we could've encouraged pair-programming to ensure better code quality and better communication among team members. |

Lessons Learned