

Reflection
Colisee by Team A

Russley Shaw rfswf5@mst.edu

December 9, 2016

For the Fall 2016 academic semester, I was the project lead for *Team A*: the team responsible for the development of Colisee. The goal of Colisee was to serve as an initial implementation of MST ACM SIG-Game's Arena component. Specifically, it was our attempt at a system that would gather competitor code, build it in a secure environment, schedule matches to be played against individual clients and then play those clients in a secure environment.

Our system, Colisee, is classified as an *Open Source Contribution*. Our code is available publicly on the code hosting site, *GitHub*, and our client, *SIG-Game*, is also a primarily open source student organization. It is a contribution because the Colisee product is not yet in its final state. SIG-Game will take our contribution and bring it to completion for integration into their system.

Joining *Team A* as lead, I already had some experience with software development technologies, such as git version control, GitHub code hosting, and Slack communication platform. All of these, helped our development process; however, a tool is helpful if it is used. Our code was hosted on GitHub, so interaction with git was a requirement, but there was minimal interaction with GitHub's issue system. For those who used the issue system, it was very helpful because I could see the individual progress of a feature, or be able to respond quickly to issue specific questions. One piece of functionality that GitHub does not have that would have been helpful was a measure of issue difficulty. Systems such as *Jira* offer a way to assign points to an individual task. This would have been helpful in calculating *velocity* as well as setting a group *target velocity*. This would have allowed me to keep better track of our progress and individual member's progress. Slack was another helpful tool. Similarly with GitHub, although it was not used 100 percent effectively, when it was used, it proved to be an exceptional tool. Slack is a chat platform with capabilities for file and code transfer. A member of our group would message me asking about an error and the member would be able to easily send a screenshot or snippet of code properly formatted and even syntax highlighted. This made one-on-one virtual interaction easy and painless.

Throughout this development we used a form of the Agile development process. We would meet once or twice a week to discuss progress, roadblocks, questions, comments or concerns. After the quick meeting those who were having issues would stick around and other members, including myself, would be able to help. This was effective for a while; however, due to scheduling conflicts, several members were not able to stay for long. The days that were most productive is when the entire team was in a single location for a set period of time, all working. To balance out this conflict of schedules, I tried to get the team to use Slack, but only several used it somewhat frequently.

I have learned several things from being a lead in this class. The most important of which is probably that there is a difference between a management lead and a technical lead. This semester, I unknowingly tried to act as both. This was because I was one of the few people who understood *Javascript* in our group, as well as the goals of the system. Also as technical lead, I made assumptions about the knowledge of my team members. Picking up Javascript was more difficult than I expected. Also, because this development effort was largely an investigative effort, technologies such as Docker, linting utilities, PostgreSQL, continuous integration were covered which are all far outside the scope of a the traditional Computer Science curriculum. The concept of the Arena is a difficult component to understand, even when a list of requirements are laid out. I also noticed that when the members were faced with a new technology, or language, pairing them up proved to be effective as they would be able to learn off each other.

The ability to track velocity I believe would had helped motivate individual members more by having a more clear goal of the expected work. Two issues would not have to carry the same

weight and being able to identify members who were not meeting a weekly goal could be met with individually to discuss why that goal was not met.

Next semester, I plan on taking the next level of this course, CS4097, where I hope to employ a more effective development strategy by becoming only a management lead and assigning a technical lead, tracking development velocity, encourage more/longer development group meetings and more strongly encourage Slack usage.