SCRUM – The good

This tool is certainly very interesting. By separating the project into manageable sprints, the project becomes less overwhelming by far. By separating out each task, and the ones ahead of them, members of the group can complete tasks and figure out what is next all without having to consult with the group or management.

Additionally, a fairly severe problem in software development seems to be that quite a few projects go through their lifecycles, end up being canceled, and in the end have no working code to show for the efforts made. The scrum lifecycle combats this by offering the promise of a working prototype at the end of each sprint. In the situation where the project was canceled, there would presumably be at least part of a useful project to the customer, so they wouldn’t have to see a complete waste of their investment. Because of this, and because these senior projects have a reputation of not being completed, I believe scrum is a good choice for these classes.

Being able to drag items into the backlog and being able to check tasks off as they are completed is incredibly satisfactory as well.

We currently find ourselves finishing up the third sprint of our journey. We find ourselves falling slightly behind the pace we’ve expected, but we’d like to attribute that to the time spent to learn this new system, and a lack of a full understanding of how to implement it. We now have a more thorough understanding however, and in Fall Quarter we expect to hit the group running to finish this project.

The bad

By trying to focus on the development process, it seems like our designs were not very well established before we attempted to write the code. Because of this, things became scattered and confusing. Instead of devoting all of our time into the design of our project, we were presented with two tasks, which each got half of our attention. This could be attributed possibly to the confusion caused by being new developers, new designers, learning new technology, and to the time it took to learn the SCRUM system as a whole.

Also note that the part of the scrum cycle where developers can grab the next story really doesn’t apply well to the classroom. In a work environment, you are paid per hour of work, so obviously you’d want to grab the next item, and the faster you move, the more you impress your boss. Although the incentive to be impressive remains, in a student setting, we simply do not have an incentive to pour time into this project as opposed to the three other ones we are currently having to do in another class. In a student mindset, it’s more of an incentive to finish the current sprint so that we can make room for the other projects that we have to do. We argue that Scrum works best when the only project each team member has to focus on is the one in question.

I cannot say that we would have preferred the waterfall system, as none of us have completed a full cycle of that implementation either.