

Personal Reflection Essay

John Carlyle

December 9, 2013

The Toilet Mapper group was an extremely learning positive experience. Each team member was individually motivated and had a slightly different background which contributed to an efficient, driven team. The team was small enough that very little oversight was necessary. For the most part, tasks were self-assigned and carried out without any oversight except for the final merging of code. Liad and Michael are familiar with making apps for both Android and iOS, when the time comes to port our website to apps they will be the people to do so. To make up for not contributing as much to the website portion of the project they both became our Google API experts. Morgan and Myself were the resident experts in Django and web development in general. Ace and Jeremy were enrolled in the databases class and helped quite a bit with database design, test writing, and general QA and bug fixing. Luckily all our team members had the same goal in this class: mainly to get an A and to make a good product we could be proud of, so there were no slackers or anyone unwilling to work hard. This has definitely could have been a major problem.

Interpersonal disputes didnt arise in our short three months together. Each member asked for help early on if they didnt understand something, and everyone accomplished their tasks in a timely manner without any intervention by anyone else. That is not to say everything was perfect from the first sprint, there were a few disruptions which we learned to resolve and adjusted our workflow accordingly. For example, during the first sprint we didn't have a mechanism in place for tracking who was doing what, and because of that, at the end of the sprint it turned out that Liad and Michael had written the exact same Google API code for two pages. We changed our groups workflow in sprint 2 to use githubs bug tracker tool. This tool allowed us to submit all user story tasks as bugs labeled enhancement. Each person could then

assign themselves the tasks they wanted and any leftovers could be assigned to people who completed their self-assigned workload the fastest. Most problems were solved by using the bug tracker but there was still an issue of double de-bugging - that is, where some people would use the bug tracker but then forget to check off that they had fixed said bug, which other people would still think was broken and try to fix. For example, this issue popped up when Ace, Morgan, and Liad worked out our query system for pulling only nearby toilets from the database. In the meantime Michael, Jeremy, and myself were in an adjacent room working on the profanity filter and map. Unfortunately, Ace, Morgan and Liad forgot to mark the task as completed on github. The next day I decided to get the hardest feature out of the way by implementing all the same things they had done the previous day. Needless to say, the code I spent a whole day writing was entirely useless. After that we were much more disciplined in our usage of the github bug tracker. The last week before the final sprint was different, we had to break into three teams to accomplish three tasks: Morgan and Myself closed as many bugs as possible, Liad (a former DJ) made our final presentation slideshow, and everyone else prepared the deliverables.

Our group had the largest problem with the course deliverables. Our team as a unit did not need weekly checkups, as this only hindered our progress and distracted us from the work at hand. The same can be said of the sprint plan/results and scrum board to varying degrees. The physical scrum board was a huge hindrance, even after the quarter is winding down I still lament the amount of time I spent tediously filling out slips of paper and carefully keeping the state of the scrum board synced with our bug reporting tool. Going forward we are looking forward to increased productivity due to the decreased overhead on all our tasks. I would estimate that for every three hours I spent coding, reviewing code, or refactoring, about a half an hour was spent writing reports or moving things around on the physical scrum board. Considering how hard we pushed ourselves during this project I find that ratio to be extremely inhibitive to productivity.

As we proceed with the project, we will use fewer self reporting tools and simply use scrum meetings between sprints or whenever is convenient. These meetings will not be in person and will use some online tools, (as yet to be determined) since Ace will be moving to Washington to work at Microsoft and Liad will be living in San Jose and working at EMC. This class prepared me for working on a software development team by forcing me to work in a team setting. The rest of the class simulated having lots of

overhead, but as research states, team reports are a really bad idea. The overhead typically found in the industry helped me understand what kind of questions I should be asking interviewers during job interviews. This includes: How do you how does your company or team handle interpersonal disputes? What kind of reporting tools do you use? How difficult is the bug reporting process in your company?

Overall this class gave me a realistic picture of what working in a larger group is like. I personally had never worked with a team of more than two people before. I am in the semi-unique position of having been programming for much longer than most of my peers (10 years instead of 2-3) so I have yet to experience being in a group where I am a junior programmer. It would be a valuable experience to learn how to act and what the mindset is of a junior programmer, and something I could probably learn if I did an internship. In closing I would like to say that I feel the crowdgrader experiment was a failure, and my peers are not capable, qualified, or willing to teach me how to write. I got a bunch of random grades that have little correalation with the quality of work I put forth. Perhaps crowdgrader is more suited for a topic like algebra where an answer key can be provided and answers are stricty correct or not correct.