

Final Exercise

John Foley

6 April 2014

Question 1

I will be talking about my two teams separately here. First, I will discuss my small group team before the merge, and then the subgroup that came together after the merge. The first group consisted of myself, Caitlyn, Daniel, Bryce, Josh, and Cody. The second group I worked with consisted of myself, Brandon, Anna, and Nick.

The first group worked primarily from the initiation of dialogue with the client to user stories. Our main purpose was to elicit domain details, requirements, and subsequent user stories from predetermined questions. Our closest way of matching proper software engineering practices and techniques was in the form of thoroughness and traceability. I feel that we kept track of details about questions well enough to form strong requirements, which the entire system was modeled after. We were very new to the application domain, and attempted to compensate by taking notes, discussing wording of questions in length amongst ourselves, and clearly tracing each phase to the previous phase. This way we could remember our reasoning, and adapt the project all the way to the foundation if necessary.

The second group worked primarily from the creation of user stories to the end of the application life as we know it. We worked very well together, and I'd say the one way that the team matched proper software engineering practices most was using Test Driven Development to shape the end product for a user story. We would either pick a user story for the iteration, or be given it by another group, and then immediately talk it through and create an appropriate test. This test was built and given values that most closely matched what the user story was meant to achieve. This guided us in the actual implementation and helped us to realize the power and all knowing

consistency of the green bar.

Question 2

I would say that the poorest way that my group matched proper software engineering practices was our lack of starting ultra simple and systematically incrementing forward. I know that this is in part to our lack of understanding of the client domain, and thus requiring that we think deeply about the project before writing any code. This was obviously necessary and not necessarily wrong, but then after the project team was formed, the team went into a frenzy and started implementing carelessly.

The pitfall was our sudden frenzy. We should have taken a moment to step back and ask what user stories would provide a super minimal, bare basic system to start from. Something so simple and quick that we could have something to provide green bars. A project manager or architect should have been chosen to help guide the team as well; someone whose job was to take a step back and look at how the system is pieced together and not be bothered with specific implementation. From the simple system, we could work forward to accomplish user stories and stay systematic, instead of trying to tackle an entire problem (i.e. write the entire transformation immediately).

Question 3

I would like to start with saying that the way the class was implemented, after some reflection, was a fabulous learning experience. We learned the basics of software engineering and the practices from the books and discussed proper methods of implementation while also working on understanding the user domain and planning a system. All of this is critical in software engineering, especially when there really isn't much of a foundation. I was stressed about the lack of process during implementation as said above, but after the dust settled I truly understand the merit and process much, much better.

If I had to change something, it would be that the frenzy should have been interrupted and forced to stop. I understand agile methodologies much better now and why they're in place, so now I want practice in actually implementing them. I do use them in other, unrelated projects so I have an idea of how to work through it, but practice in a supervised environment with a team

would be very beneficial. The frenzy was a great learning experience, but if I could go back and do it all again, I would have stuck to a real agile environment learn implementation through success and maybe not the hard way.

Question 4

John Foley (myself): I would say that my biggest critique to myself is not maintaining concentration through group meetings. I went to and communicated meaningfully to every meeting, but as the group discussions would go on for hours I would lose focus and stop participating. To remedy this, I should have attempted to put the group on a better track and make better use of our time. My strengths are organizing, code contributions, and leading communication between teams. Weaknesses include consistent participation.

Brandon Medina: Brandon came to every meeting with some degree of punctuality and contributed meaningfully to group discussions. Unfortunately the group was too massive, even within our subgroup, to contribute fairly to code during actual coding sessions, but he made up for it with having a presence to bounce ideas off of and spuratic code reviews. My biggest critique of Brandon was his lack of code and drive to contribute. Strengths includ involvement and quality insights, and weaknesses include participation and punctuality.

Nick Moser: Nick served as a consistent leader for the group and ensured that we were always on track. He contributed to the project in terms of code and ideas, and mainted a presence at all meetings. I liked how he handled different ideas and the ability to make a decision and then back it up. It helped further the data model's group progress, and I really appreciated that. I don't have any significant critiques of Nick's performance. Strengths include code contribution, project leadership, and quality of communication. Weaknesses include consistent participation with group discussions.

Anna Goodloe: Anna came to all group meetings on time and helped organize meeting times between subgroup members. She and I pair programmed together most of the time, and thus contributed code and helped watch my code for errors. She was always happy to be there and contributed ideas to group discussions. I would say my biggest critique was her lack of new insights into the project and willingness to proactively understand what other teams were implementing.

Strengths include participation and pair programming accompaniment, weaknesses include meaningful new ideas and quality of code.

Chip Thien: I worked with Chip a lot to design the interface between the data model and the transformation engine. He was great to work with and communicated effectively to keep things moving forward. I know that he contributed much to the logic and code of the engine, and researched with me for different API classes that we could use. I have no major critiques of his performance. Strengths include activity in the group and communication between teams, weaknesses include contribution of new ideas to group discussions.

Victor Jankov: I worked with Victor to design the interface between the data model and simulation engine. He was very excited to work out the simulation and knew the system very well. I could tell he was well researched and prepared for our discussions and came to every meeting, even the data model ones. He was willing to contribute code to solve any problem or adaptation we needed, so I have no major performance critiques. Strengths include activity and drive to complete work well and timely, weaknesses include more general system knowledge, such as how other teams were implementing their design.

Question 5

I really enjoyed this semester. I knew this class would do a good job of teaching the different software engineering techniques and gain a lot of valuable insight into the industry. I feel prepared to work at Cerner this summer and the correct way to go about designing a web application. I just started working ITS Web, and impressed my supervisor by organizing different tasks that he wanted me to work on for a new Chumby application in an agile way and then using git to organize my work.

I was really stressed by the unorganization and what I thought was a loss of any process in implementation in the last few weeks, but after class today (our final meeting time) I realized that you practically planned it. I realized during our class discussion that it truly made me see the merit of having a good process and how our project clearly illustrates everything that can go wrong. I really appreciate going through all of that now because I'll be that much less likely to put myself through it the next time!

My only suggestion is that you maybe put an end to our suffering at some point. I would have liked to practice methodologies and processes under a more supervised environment (one that I'm not being paid for) so that I can be critiqued and given suggestions. Other than that, I had a great semester.