Throughout the course we took on every role on a scrum team, the product owner, scrum master, tester, and even developer. I learned quite a lot during this class, and it opened my eyes to how important the other roles of team members are. Being a current software developer myself, I never knew how in-depth the roles of a product tester and product owner are. It really amazed me how we learned that they both take time to scrub the backlog of redundant issues. On top of doing their daily testing, and creation of stories. It really did give me a new found respect for my current team. In the day to day job, we tend to only focus on our tasks. We never really consider how everyone else’s plate is as well. I’m very lucky to not have had to work in a waterfall environment very long and am glad my company made the switch. Speaking from experience and this course. Each role greatly differed than an agile way of planning and in my opinion suffered way more in the waterfall approach. Scrum masters don’t really exist in the waterfall method, so there’s no one to really keep you on track of what your goals are. On top of the product owner not being real hands on. In agile, everyone works together including the product owner. The developing in a waterfall method also seemed pretty hectic, having no room to plan for emergency stories. On top of delivering a finished product that might not even be exactly what the client wants. Agile gives everyone a lot more breathing room, to handle new client expectations on top of bugs that may pop up. It also gives the testers more time for QA on stories, and dedicated regression time. I would honestly choose an agile approach every single time over waterfall. Not just because of my personal experience, but from everything I’ve learned. There is so much that goes into planning, and tools for you to succeed in Agile. From sprint ceremonies, to user stories to even having the time to cross pollinate your skills. The tools we got to research were amazing, again I’ve been using agile tools as a developer. I just never knew there were so many interesting ones out there to make even meetings fun. One I ended up researching was for scoring sprint stories. It was called card poker; it was an innovative app that made even sprint grooming fun. You got to learn with your team, have a fun time in a meeting that are usually boring. On top of grooming your sprint and backlog for success, even if the story may not go into the sprint. It’s great to have something already groomed that you can pull on just in case you have free time at the end of the sprint. I also investigated a few sprint planning story boards that went into detail of each sprint. Showing how far along a story is makes everyone’s job easier. You won’t have your manager on your back asking where a feature is, they can simply see how it’s progressing on the story board. On top of being able to see who’s worked in an area of code, you can already plan for help before getting too far into a project you might not understand. The communication in this alone helps everyone not get as frustrated with the project and lets them know help is easily ready. On top of the scrum ceremonies, the stand up allows you to speak up and let everyone know you might be stuck, might need help. Or even let the product owner or tester know you need more information. The face to face is nice, as sometimes emails can easily be missed.

In this course, I learned just how hard it is to create user stories and give a lot of details. This is something I’m terrible with, and something I know I need to improve on. In some cases, even developers end up writing up story bugs. It’s important to gather as much details as possible, where the area effected is and what else might be affected if touching the code in that area. The artifacts we created helped with that, they really made you think of what’s important in a project along with every little detail that goes into a user story. Is this something that would be considered an epic? Is this a small or medium task? Could I possibly break this story up into several different stories to tackle each area? These are all things I learned when creating each of my stories. The thinking of everything that goes into each little detail of something as simple as a search bar. What the client may want, the whole design, how important the feature is to them. It was crazy to see it from every side at once. Going back to the tools, we researched they made it look easy to split stories and track what type of bug or issue the story was. On top of the priority, being able to mark a story with a high priority allows the team to know we need to get this into the current sprint or next one. Or if a major bug was found by a client or QA, hosting the daily stand up or even the grooming meetings allows for someone to speak up. Let others know there is an issue that needs hot fixed. Or even taken care of by the end of the sprint, this communication is vital and helps with the customer not finding the issue. As we all know when a customer finds an issue, it never ends pretty and sometimes even ends with the C levels being upset. Agile is also great because of this, being able to account for set backs and plan accordingly is great. While also knowing how much time each person has per week, in the tool we use we can assign a developer a set of work hours per day per week period. We then assign each task an hour rate. We also account for meeting times on top of general meetings and round up. On average we give each developer 6.5 hours of work time a day. Having this allotted time is so important, because it allows us to not go over our allotted time. While planning on a time-based method like this, even if you overquote which is mostly the case. It gives you extra time for a task that you might have underquoted. With the grooming meetings that happen weekly, there is also pre-groomed stories that developers can take. Virtually, there is always something to do and you can always bring in extra stories into the sprint as a bonus. Or even begin working on something in a future development branch. Being able to communicate is extremely important again, as we all know what each has on their plate with the tools we use.

Before this class I’ve spent quite a few years as a technical support representative along with being a Systems Administrator. My field has always been computer science, and I’ve always admired software development. In the last 2 years, I got a chance to prove myself and moved into a development role and love it. However, the company I’ve been working for was outdated in their methodology. Which is a little funny, seeing as the Agile method really grew in St. Pete Florida where my job is located. When getting into the position, we were waterfall based. Everyone basically had this final product we were working on, had tasks that we’d take out of the backlog and then release monthly. No planning just gets the work done. If you need help, ask someone if they have the time. During this time, we had a lot of people leave and the tech director knew something had to change. He started researching and learned about agile, he also decided he would become our scrum master. This was a new concept to everyone in the building, but we got to learn along with him while he got his certification. Though we have come a long way as a company now, this class shows me just how much better the process can really get. It also showed me that there’s still more I can approve on noting wise as a developer. I needed this course to give me the insight of a tester. So, when I do finish a story, I can explain better on what was changed. What was affected and should an issue arise later around that portion of code. Possibly help another developer solve an issue a lot easier. Like I also previously said, it gave me a new found respect for my coworkers. I honestly didn’t know they did as much as they do. I also even went the extra mile to ask one of the testers at my work on the process of grooming the backlog. It’s a lot of tedious work, and in some cases, they’re testing issues that have already been fixed. However, while that is something that is extremely important, I also learned that being a developer is something I do want to keep being. While I may have learned quite a bit about the other positions. I don’t think I like any of them as much as being a developer. The only other position I would consider, if I had to would be that of the scrum master. I know the scrum master at work well as he was my previous boss in technical support. I really admire everything he does, and what I’ve learned about in this class about the role. Getting everyone organized, getting the whole process down and watching the building of the product becoming more and more efficient each sprint seems like a very rewarding job. For the least favorite position, I would say that the product owner would be the worst. It takes a very special person to be able to talk with a client, take unrealistic standards and basically bargain with them for a realistic project. All while making sure it is still something the client will love and providing support knowledge to the team. I believe we also went over how the product owner will even groom the backlog and create stories. Product tester also didn’t suit my fancy, I knew their job was tedious before. However, after learning of everything they do I can guarantee I’d be really bored in that position. All in all, software development is the most rewarding to me. Knowing I created something in this world that thousands of people use is super rewarding. I get an amazing amount of pride when I see a client post about a new feature I coded. How they loved it and how it makes their life easier. Development is also challenging which I love, everyday is a new puzzle you might have to solve. You’re constantly learning in the field and with new updates to the IDE every so often, new functions are created to make your job easier. Or in some cases, allow you to create functionality you could not do before. You’re a pioneer, and no matter what your skill level. Sometimes you’re creating something or learning something no one in the building has done yet. You constantly are evolving and it’s so rewarding, and sometimes you do fail. Yet, being on a team that encourages you and let you know it’s ok. To keep pushing you is what makes it all worth it. I and my team strive for an environment where everyone’s input is valued. In every meeting, we discuss IS this the best way to go about something. If not speak up and we will talk about it. Everyone strives to get better, no matter what role you take on the team. As for professionally, I’m extremely proud of where I am now working for the newest and best POS cloud software for pawn shops. I’m so happy to be the innovator of new software that these clients can enjoy. Making their lives easier and creating this product has been so rewarding. I’m not sure where I’d eventually like to end up, however I do know if I can constantly learn. On top of creating software that benefits the world, I will be happy and know I chose the right path.