# CS 250 Final Project: Sprint Review and Retrospective

Name: Deny DeLatorre

Course: CS 250 - Software Development Life Cycle

Date: 10/19/2025

## Sprint Review and Retrospective

So, through this course, I’ve gone through almost every role for a Scrum team while doing assignments and projects. Although throughout this I wasn’t on an actual team, I would like to say it gave me an idea of what it’s like to plan the software product in sprints. So this reflection will cover what went well and what could’ve been better, and how using Agile makes everything smoother and flexible than other methods like waterfall.

## Applying Roles

As a developer, the main goal was to focus on writing code and thinking about what features work and which ones don’t. Now, as a tester, the main goal was to find the bugs in the code and make sure the user stories met the criteria. As Product Owner, I had to practice being the user, deciding what should be important to do first and then last. Finally, as the Scrum Owner, I had to basically guide the process as well as organize the tasks for each phase. Acting in these roles taught me how much teamwork and communication there was in an Agile workplace.

## Completing User Stories

When I started working with Agile, I noticed that user stories made everything easier to follow. Instead of trying to build the whole thing at once, I just focused on one story at a time, like small steps that actually led somewhere. Each one had a purpose, and finishing them felt like progress instead of stress. In past classes, like when I worked on other projects, I used to just rush through everything, but with this setup, I could plan things better. It made more sense to complete one task and move on, instead of juggling too much at once. That’s something I’ll definitely keep doing in the future.

## Handling Interruptions

One of the best things about using Agile was how it handled changes. There were times when something in the assignment directions changed or I misunderstood something, but instead of starting all over, I just adjusted what I was doing and kept going. It honestly made things less frustrating. If I had been doing it in a Waterfall style, I probably would’ve been stuck or had to redo a bunch of stuff. Agile kind of teaches you to go with the flow, and that really helped when things didn’t go as planned.

## Communication

Even though we didn’t actually work in teams, I still tried to communicate like we would in a real Agile environment. I wrote notes to myself about what I was finishing, what I needed to do next, and what was getting in my way. I’d sometimes imagine how I’d write to a team, like, “Hey, I just tested this feature, but it needs one more fix before we move on.” Doing that kept me organized and made me think about how much clear communication matters in real development.

## Organizational Tools

We didn’t use official Agile tools like Jira or Trello in this class, but I used my own little system. I kept lists, notes, and small checkboxes for every part of the sprint. It made me feel like I had my own board to track progress. That small habit helped me stay focused and not forget things between assignments. Even though it wasn’t fancy, it worked, and I can see how using real tools like that in a real job would make teamwork even smoother.

## Evaluating Agile Process

Overall, Agile really worked for this class. It made everything more organized and easier to adjust when things changed. The best part was that it kept me from getting overwhelmed — I just worked on one thing at a time and always knew where I was in the process. The only downside was not having an actual team to bounce ideas off of, but even without that, it showed me how strong teamwork and communication could be if it were real. Agile fit the style of these class projects perfectly because things were always moving and improving.

## Agile vs Waterfall Reflection

If this class was done in a Waterfall way, it honestly would’ve been way harder. Everything would’ve had to be planned out from the start, and changing anything halfway through would’ve been a pain. Agile made it easier to take things one step at a time and adjust when I needed to. It felt more realistic because that’s just how things go — plans change. I liked that I didn’t have to have every little thing perfect before moving on, I could just fix it as I learned. That’s what made Agile way more fitting for this class and the kind of work we did.

## Conclusion

Overall, I really liked this project and the class itself. I got a better idea of how real developers work and how much Agile actually helps keep things on track. I also liked getting to see how each role connects and how flexible everything can be. Even though it was all individual, it still felt like I was part of something that worked together. It wasn’t just about finishing assignments — it actually taught me how to think more like a developer.