

Sprint 3 Reflection - November 2nd, 2019

This sprint focused on additional tasks to improve the UI through front end work and expanding the backend database work. We wanted to connect the two and be able to produce a login, list of users, and ability to send messages directly.

Workflow / Process

Our team worked very well in dividing up the workload for tickets. We created stories for each ticket and assigned story points. We used the Fibonacci sequence to effectively grade complexity across the tickets. We effectively had above 80% code coverage.

Code Quality

Although code coverage was good due to testing and we were able to reduce smells to 1 (this one smell was due to Tomcat requiring a line of code that SonarQube considered deprecated), I think our stylization and structure was a bit confusing. There are extraneous classes that were created and could have been deleted. Also, some refactoring is needed to keep the code clean and simple for one or two classes. Some team members were using different environment settings so that led to some code having different indentation or using different formats. We are still facing the issue of different Java JDKs which lead to dependency issues. We need to resolve this so that everyone is using the same environment and the same code.

Scrum Master

The scrum master was pretty much MIA. I think there was very little communication and I don't think he knew enough about Agile to effectively provide us with enough feedback and direction. Although half the battle is knowing and trying to learn, he also needed to communicate better, which is a soft skill that is required for this role. No progress was made because the scrum master decided to take away a lot of people's tickets into his column. This turned the scrum progress more into a Kan Ban process where the team members have one ticket each. He was expecting to dole out the tickets when everyone was done. However, this was never communicated, so we were a bit confused. I think the scrum master role should be removed, because an ineffective scrum master can really deter a team from work. We don't teach this methodology in class and just expect untrained students to take this role.

Technology

I think we are picking up speed on all our technologies. While the initial two sprints were quite difficult, the team members that were new to this tech stack finally understood how the tech

worked. I am not 100% sure if people are actually using SonarQube locally, but we are submitting branches with less or no smells so that is a good sign.

Teaming

I think Dian and I contributed the most and helped Alice out with her tickets. Alice had a learning curve, but that didn't deter her from finishing her tickets. The only person that did very little was David. I don't think he contributed much. He communicated very little and was oppositional even though we were trying to mentor him (considering his lack of knowledge on many things). He took the scrum master role as more of a leadership role and decided to turn scrum into a Kan Ban style where he would take away all our tickets without communicating any of this with anyone.

Improvement

The team needs to work on tickets at the beginning of the sprint. Historically, a couple of team members wait until week two to start work. This is not working out and leads to last minute tickets two hours before sprint deadline.

Accomplishments

I think we are on a great path to polishing the app and making it look professional. We were able to setup the database and connect the front end, which is great progress.

Developer

We met as team to consider the sprint backlog. In this sense, I was able to contribute to front end ideas and feasibility since this is my area of expertise. I did raise issues with how tickets are groomed and asked that everyone groom the tickets the first time they receive them to raise any issues immediately. Unfortunately, this fell on deaf ears for one team member and they ended up passing the ticket back to everyone. I was able to help Alice by collaborating on javascript tickets together.

A lot of the work was UI and front end work, which I currently do for my actual job. The tickets depended on the web socket to provide JSON objects based on the Database columns, so I worked directly with Dian. Alice and I actually worked on a ticket together since she has some difficulty and I knew Javascript. I also worked on mocks and designed how the app will look. These were all straight forward tasks and easy to me since I have background knowledge on how web apps work. The front-end code was not being checked by SonarQube since this

wasn't a requirement; however, I still committed to quality of code and readability since I know that my teammates will be working on this code in the future.

I definitely committed to most meetings and all of the stand ups. There was only one meeting I could not attend due to extenuating circumstance. I actively communicated with others and mentored them on front end code as needed. I also communicated with people if issues arise. AWS was about to shutdown our instance, so I actively communicated with the team and the AWS customer service rep on what we could do to effectively migrate our instance over to another AWS account with credits.

Team Member Reflections

Good

We collaborated very well on tickets since they had dependencies between the front end and database and required lots of communication and even some co-coding.

Average

Dian, Alice and I all completed our tickets in a timely manner.

Bad

David was completely MIA and not completing his stand up or tickets and did not communicate very well during this sprint. When the team comes together for decisions, he was oppositional when making many decisions, even though everyone else is in agreement. I think he needs to understand that he must get on board with the team. I believe he doesn't know what he doesn't know, but is argumentative and oppositional when being mentored and asked to do the correct thing (even when you explain to him why it is correct). This is becoming taxing, because decisions take longer to make and more work is being placed on other people as a result. Worse, he doesn't see the impact of his actions and therefore cannot correct them.

Rating		Comment
Tanner	7	I mentored Alice in front end work and worked with Dian to understand how the database connected to the front end. I worked on migrating our AWS instance with Dian and interfaced with the AWS CR when needed.
Dian	7	He worked tirelessly on the database and a lot on the REST services. I believe the core of the application is a lot of his work.
Alice	5	Completed her tickets in a timely manner.

Rating		Comment
David	0, maybe 1	He completed one ticket with subpar code. I don't need to reiterate what I wrote above in the bad section. Lots has been said. He took everyone's ticket and decided not to do any of the six tickets that he reassigned to himself. So then he reassigned them back to everyone two days before the end of the sprint. I had cover lost time. For his role as Scrum master, he did absolutely nothing. Zero communication.