Scrum Master Report for Team B-16, written by Andrew Nguyen-Tran

Summary

Throughout the semester, our team partook on a transformative journey in CS 3240, developing a fog-based exploration game for users to discover new places across UVA grounds. From the beginning of Sprint 1, where we grappled with conceptualizing our project and establishing workflows, to the technical and interpersonal challenges we overcame, our progression was marked by continuous improvement and development. Key achievements included mastering Django, integrating the Google Maps API, and deploying our application to Heroku, all while enhancing our Agile methodologies. The team's chemistry and effective communication played a pivotal role, especially in addressing technical challenges like migration errors and balancing product development with quality assurance. Our adaptability was tested, but our proactive strategies, such as prioritization matrices and team training sessions, ensured smooth progression and problem-solving. Our journey resulted in a functional, quality-assured web application: a testament to our hard work, collaboration, and the invaluable skills we gained, preparing us for future endeavors in both academic and professional settings.

Report

Over the course of the semester, our team has embarked on a journey of continuous improvement and development. This report consists of the experiences, challenges, and achievements that we encountered throughout our project — a fog-based exploration game to explore around grounds to find hidden gem locations — in CS 3240, offering a holistic overview of our journey.

First, I will provide an overview of our sprints throughout the semester. Sprint 1 marked the commencement of our project. Initial tasks involved setting up the environment, establishing a workflow, and assigning roles to each team member. The team showed great amounts of enthusiasm, but we were already encountering challenges such as what we wanted our project to be (what the application should entail, known locations vs. hidden locations, etc.) and figuring out how we were going to approach requirements elicitation. In Sprint 2, we saw an improvement in team dynamics and chemistry. We were able to successfully carry out our requirements elicitation methods; we received massive amounts of feedback regarding our application. Consequently, our team had a little bit of trouble refining our final set of requirements. Furthermore, we were challenged in converting these plain-language requirements into actual software requirements and features that we could reference as developers. Nonetheless, with extra effort and teamwork, we were able to properly create our requirements. Next, Sprint 3 brought us the most technical challenges compared to other sprint reports. These challenges included creating the Google Login and getting authentication to work and connecting our web page to Heroku. Most of us were new to Django and overall creating web applications, so we struggled with learning these new topics and applying them. Since another team member and I had previous experience, we offered to meet up with the rest of the team and hold a seminar where we would teach them Django and overall web application techniques. I found that training session to be immensely rewarding and worthwhile as not only did it help the team understand what was going on, but it also helped us with future sprints. Rather than our teammates struggling to understand what was happening in the code, they were able to comprehend it and make changes on their own.

As a result, the second half of the project was much more smooth as everyone became more comfortable with Django and our project was seeing progress. Sprint 4 was when we started seeing this progress; it was at this stage that our team started getting really excited due to how promising our application was becoming. In this sprint, we incorporated the Google Maps API and got the GitHub Actions CI to be operational as we created a few tests. Surprisingly, these tasks were not as technically challenging, most likely due to our team's effective communication and ability to learn and ask for help. However, a challenge that we did have was linking all the features and components to each other to create a functioning website. Sprint 5 was relatively simple as there were no specific tasks to complete, but rather continuing to improve our application and making it more functional. We implemented new features such as creating a leaderboard, making an onboarding screen, and users having the ability to create submissions for administrators to approve. In this sprint, we encountered an unforeseen error that our group spent a lot of time trying to resolve. There were issues with our migrations regarding our database, so to resolve it we had to delete our SQLite database and start from scratch. This error bug taught us that we had to be prepared for anything to happen and learn to be flexible. Lastly, Sprint 6 was our last sprint and we prepared for beta testing. For this sprint, we just focused on mainly making sure that our application was functional and did not crash. In order to do so, we did an immense amount of testing and focused mainly on developing the user flow of our application so that other students (and users in general) can use it from beginning to end smoothly. We worked on creating new features and perfecting our existing ones. This included creating a navigation bar and logo and improving the map feature of our application. In general, throughout these sprints, our team worked together very well. The enthusiasm that we showed from our very first sprint was a strong indicator that we were going to have a great time as a team. However, I believe that the biggest factor was the emphasis that we had on how important this project was to our grade and that teamwork would be the biggest reason for our success.

Now that I provided a thorough overview of our team's progress throughout the sprints, I will now be condensing it and discussing our team's achievements and challenges. Throughout the sprints, our team not only developed a functional product but also grew in our understanding of Agile practices. Some achievements included creating a web application using Django and deploying it to Heroku, successfully implementing Agile methodologies, and achieving significant product development milestones. However, our most notable achievement was enhancing our team collaboration and communication. Without a doubt, our team's success was mainly due to our ability to work with each other well. We were able to implement changes and work together to make sure we carried out our roles, but more importantly, we overcame technical and interpersonal challenges. Our technical challenges included Django fundamentals, migration errors, Heroku errors, and overall lack of understanding in creating web applications. However, as mentioned earlier, these were easy to resolve with some teamwork and empathy. Another technical challenge that our team encountered was balancing product development and quality assurance. While it was important to continuously develop our application by implementing/improving new features, it was crucial to ensure that our application was of high quality. At the end of the day, it didn't matter what cool features our application had if it wasn't functional. To ensure that our group was balancing product development and quality assurance, we prioritized our tasks. A tactic we used to do this was prioritization matrices, which is a method that uses a matrix to plot requirements based on multiple factors that determine their importance/urgency. We did this by synthesizing information from previous steps and prioritizing the requirements. Thus, our outcome yielded a visual representation of which requirements should be tackled first. Overall, our team's ability to work together and understand what needed to be done were crucial to overcoming our technical challenges.

As for interpersonal challenges, I can happily say that our team didn't really face any challenges. Other than not being able to agree on some things — such as what type of application we wanted and how to pursue requirements elicitation — in the beginning stages of the project, our team got along really well. Yes,

occasionally some members (including myself) inevitably took a while to respond to the group chat, but it was never significant enough to truly hinder our team's performance. We understood that everyone was busy and always had their own things going on, so texts could easily slip by them. A simple solution that we discovered was sending reminder texts, in other words, double texting. Doing so helped our team members remember and feel the need to respond. Another important aspect to our team's communication that we discovered early on was what communication practices worked and what didn't for our team. We came to the verdict that text messages were best. Other platforms (even Discord) were not optimal as some team members didn't check them frequently enough. Through mutual understanding and adapting our communication methods, we effectively navigated minor challenges and maintained a productive team dynamic throughout the project.

As for advice to give to teams next semester, my single most important piece of advice is teamwork. Understanding the significance of collaboration and communication for big group projects such as this is truly crucial to a group's success. By emphasizing the importance of this at my group's first sprint meeting, we all started off on the same page. If really needed, threatening to give teammates bad evaluations if they don't collaborate well could help encourage them to work with the team (although my team never had to go to this extent). Another piece of advice is seeing potential issues early on and resolving them as soon as possible before they accumulate. The more that issues accumulate, the less that anybody wants to revolve them, which will contribute to the downfall of a team project. An example of how our team promptly took care of an issue we saw was in Sprint 3. We started seeing how the lack of experience some team members had with web applications was impacting our project, so we quickly resolved this by holding a seminar where they could develop a fundamental understanding. This tremendously helped our team's efficiency and success later on when members quickly grasped what tasks we had to do and were able to complete them smoothly. Lastly, another piece of advice that I have is to be flexible and don't procrastinate. This was evident in our Sprint 5 when we encountered migration errors that took days to resolve. Had we done our work last minute, we would not have been able to fix the errors by the due date of the sprint report. Additionally, our team's flexibility and fast-acting mindsets were crucial in getting those resolved in time. In general, each team is different and should figure out their own ways of being functional. However, in terms of our team, the staff should know that our team worked extremely hard on this project and believe that we went over the top with our application.

Overall, I believe that our team worked very well together and we were able to build team chemistry relatively quickly, which contributed to our overall success. By working together to achieve wonderful things, and more importantly overcome technical and interpersonal challenges, we were able to create a web application that I can confidently say we are proud of. However, we also learned many new skills (whether technical or soft) that can apply to our future courses and jobs.