Steven Carik

1. Worked well   
 a. When we were all at meetings I feel we communicated quite well what we needed to do   
 b. I believe that we were efficient in helping each other when someone hit a roadblock.  
 c. When we encountered problems we were able to get through them in a fairly reasonable amount of time  
 d. I think we separately the work quite evenly among all members  
 e. Meeting and working in the same area was when we were most efficient  
2. Didn’t work well  
 a. The way we split up the work isn’t the most efficient  
 b. We all need to make it to every meeting  
 c. Inconsistent meeting times and places  
 d. Communicating outside of working together on what each person is working on in regards to github.  
3. What have you learned  
 a. I have learned that it is incredibly important to to get together with your group on a regular basis  
 b. Making sure you come to the meetings prepared with all your work done so you can ask questions or discussion any problems you may have is worth it  
 c. I have learned a fair bit about how scrum works and the effectiveness of agile development.  
 d. I have learned the process of moving things throughout the columns so that everyone can see what is done and what needs to be done.  
4. Confused  
 a. I still don’t fully understand the role of the scrum master in some of the meetings  
 b. Or the most effective way to distribute the work

Adam:

1. What worked well?

When we actually met and worked together through challenges, our progress was very rapid. Our communication via FB Messenger worked pretty well. We helped each other out very nicely on multiple occasions.

2. What didn’t work so well?

The creation of tasks didn't go super well. It was also really difficult to rate them by percentage of a user story. The database was very tricky to deal with at first, but I think we've got it figured out now.

3. What have I learned?

We are a good team when we divide the labor up evenly and provide support to one another when someone is stuck. We need to do a better job of dividing up tasks in a meaningful way and we need to have people specialize in either front-end or back-end coding.

4. What still puzzles me?

Some of the graphical requirements seem like they are going to be a little bit difficult to implement in the future.

Erin:

1. What worked well?
   1. When we were working together for each daily scrum meeting, we tend to get a lot of work. We saw this more towards the end of the first sprint, when we had the most amount of problems and the most work done.
2. What didn’t work so well?
   1. Our meeting times did not work very well. We all had added things into our schedule which made it hard for us to meet. Also, when we did meet, at the beginning we weren’t sure how to go about things and what was most efficient for the project. Another thing was all the illnesses a couple of our group members suffered as the project was continuing. That also made it hard for us to see each other.
3. What have I learned?
   1. I have learned that we should work together more often when it comes to problems we have regarding the project. I think our best course of action is to write down any problems we have regarding the project and bring them up during our meeting because it seemed like other members had great ideas on how to solve certain problems.
4. What still puzzles me?
   1. Nothing really.

Matt:

1. What worked well?
   1. We were able to communicate via Facebook Messenger, in a relatively timely and organized manner.
2. What didn’t work well?
   1. Some members of our group have sub-par timeliness.
3. What have I learned?
   1. How to work in an agile development focused group. Not to take all of the work under my own ideas. To communicate more with my group. Not to touch other people’s water bottles.
4. What puzzles you?
   1. Everything, I don’t really remember if I did anything for this class/group.

The team’s response:

1. What worked well in the sprint?
   1. Throughout the sprint we all were able to make it to a majority of the meetings and contribute to the meeting and the project significantly once we found our footing in the language. As far as splitting up the work we tried our best to give each member the same amount of work throughout the sprint. In line with that we tried to separate the task so that each member would get to work with a different aspect of the project. Our communication during the sprint wasn’t perfect, but it still worked for us and we were able to relay messages so that others could understand what was going and so that we could plan for the next meeting. We also used facebook messaging to help control who was working on what file so that we didn’t have problems while committing. During the meetings, upon discussing what problems we were having, we collectively worked together to solve the problems we were having. Duh end bruv.
2. What didn't work so well in the sprint AND WHY?
   1. We didn’t split up our tasks efficiently. This is partly because we didn’t know which portions of the project would need direct access to the backend and which would be mainly frontend dominated.
   2. We had a difficult time keeping up with the proposed deadlines for completing tasks. This was because we were still learning scrum, we had to change to a new language since mysql wasn’t really supported with qT without having to do extra research on how to install mysql drivers, and two of our members were sick for a large duration of the sprint.
3. What the team plans to change for next sprint AND HOW (up to three things)? Why do you expect these changes to improve your performance / productivity in the next sprint?
   1. We should have one person specialize on frontend code, two people specialize on backend code, and one person be the tester. This will streamline the programming process and speed up production.
   2. Tasks should involve only frontend or backend code. They shouldn’t mix. This way members can work on tasks that are specific to their specialty.
   3. We should also create a list of problems on the github scrum board that we are facing, so we can ask the dev team for their input or other solutions. This and not being afraid to ask questions when it comes to anything with the project. This would allow members’ problems to be more readily available to the group. Hopefully this would speed of the time it takes to solve each problem.
   4. We should remember to keep daily scrum meetings to 15 minutes long. Honor the timebox. This will hopefully make us more efficient.