Sprint Retrospective Report - Team It'sNotABugIt'sAFeature

Individual Responses:

Kenzie:

1. What worked well?

-Pair programming was a great way to assure all team members had the chance to learn the languages and the structures used for the project. It also helped having other team members there when running into issues. Pair programming also helped me initially when my laptop wasn't working.

-Using Ruby and MySQL were both great choices for languages because they were easy to learn and use.

2. What didn't work so well?

-Using personal laptops was initially an issue due to errors with installing software needed to create and host our project. Now that everything is addressed, using personal laptops is very nice.

-Having issues with my laptop inhibited me from working on the project alongside my team, which, in turn, slowed down the rest of the team's velocity. This was really frustrating for all of us, especially when it interrupted meetings and when I was not able to help with tasks.

-Initially, our Daily Standup Meetings were going way too long, but this was often due to getting distracted/addressing issues with software on my laptop

3. What have I learned?

-How to use VirtualBox
-How to start writing and testing in Ruby
-How CSS works
-Bits of JQuery and JavaScript
-The importance of having a working system to work with the project
-The importance of being on time to meetings to make the most of them
-the correct timebox for Daily Standups

4. What still puzzles me?

-I need to catch up with learning syntax of Ruby and Rails; I don't feel super comfortable with either yet, so Google has been useful, as well as Matthew and Michael, who have excelled at learning these both quickly.

-JQuery and JavaScript syntax
-CSS syntax
-Understanding what we're supposed to do with auto-generated folders and files when using certain gems
-Database migrations

Michael:

1. What worked well?

- Pair programming worked exceptionally well for us. We were able to complete tasks together and learn both the language and the syntax at the same time, and both members could help each other when we got stuck.
- Also, Ruby on Rails was an excellent choice for a language - it was very easy to learn and very intuitive, and there were excellent tutorials on youtube. Our velocity was high because of this.

2. What didn't work so well?
   - I'm not nearly as good at front-end styling as other members of the group, so I need to work on that for next sprint. I think I just need to spend more time with it.
   - A couple of us were not on time to meetings all the time and that decreased the usefulness of our meeting time.
   - At the beginning of the sprint our "daily standups" were wayyyy too long and we didn't allocate time well. We fixed that toward the end.
   - We learned it is much better to have access to a unix terminal of some kind, installing Virtual Box on Haylie's and Kenzie's machines was a pain.
   - Not all of us know Rails super well so some time has to be spent catching up instead of being productive the whole time.
3. What have I learned?
   - I have learned that working with web frameworks is not as scary as I had thought originally!
   - I also learned which members of my team that I'm more or less productive with.
   - I learned how to do unit testing in Rails and I implemented many of the tests we have written.
   - I learned that any coding problem can be solved by enough time spent with google.
4. What still puzzles me?
   - Rails db interactions (db:migrate, db:reset, etc.)
   - Passing info/parameters between pages via controllers

Matthew:

1. What worked well?

Pair programming helped me quickly program things. I was able to run ideas by who I was working with and brainstorm more quickly. Making decisions on implementation choices was easier when I could ask the person right next to me. Once I got the hang of Rails, my velocity became very fast. We were also very communicative with each other- using Slack helped a lot with this. Our team gets along well.

2. What didn't work so well?

Kenzie's technical problems were a big issue that took a lot of time to fix. I wish we could've fixed them sooner. Not everyone is on the same page when it comes to knowledge of Rails. I would like the team to be at a level of understanding where we could complete any story on their own. Even though we do pair programming, our velocity could be better with a better understanding of the technologies.

3. What have I learned?

I have gained a decent understanding of Ruby, Rails, and AJAX. Rails has a lot of parts that work together, but it is very intuitive and easy to work with. The hardest part about learning AJAX was figuring out that it was the technique that met our requirements.

4. What still puzzles me?

I still have trouble with Ruby syntax sometimes. What kind of language has “unless” statements? Database migrations are difficult and I don’t have a complete understanding of how they work, but I’ve managed to create migrations for every table we need so far.

Haylie:

1. What worked well?

I really liked Pair Programming, I find it very helpful especially since both Matthew and Michael are more familiar with the language and I am not (due to the technical problems earlier, I was a little behind). Ruby is a fairly easy language to understand. Team chemistry is really awesome, we all get along very well which is pretty good considering that we meet almost every day.

2. What didn’t work so well?

The technical problems both Kenzie and I faced (though mine are minor compared to hers) I felt slowed us down a little. I know that for me personally I felt like I didn’t contribute as much as I could have because of technical problems, and I still feel rather behind in terms of knowledge about Rails. Also, our meetings tend to be considerably longer than necessary because we get distracted by one another. This certainly isn’t the worst problem to have, but after some time, it can do more harm than good. I feel that we’ve been better about sticking to the time box about that.

3. What have I learned?

While I still have a lot to learn about Ruby and Rails, I have learned some amount. Also because of my technical issues, I learned how to install VirtualBox on my machine (many thanks to Matthew), and how to use it. I’ve also learned a little bit of CSS

4. What still puzzles me?

I could still learn a *lot* more about Ruby, I don’t feel completely comfortable to code by myself.

Group Report

What worked well this sprint?

- Pair Programming
    - This worked particularly well because we were able to brainstorm ideas more quickly than if we worked alone. It sped up the learning process because we could have two people googling something at the same time, so we fixed technical issues more quickly.

- Ruby on Rails
    - RoR is a very intuitive and high-level language, which helped immensely when we were learning it. There were quite a few very helpful tutorials online that allowed us to learn on our own, and our meeting time was better spent because of this. This way, instead of having to learn a new concept or functionality from scratch during a meeting, if one of us was confused we could talk to the other members of the group for clarification during meetings. This improved our velocity.
- Velocity
    - We (Matthew) were able to complete tasks pretty quickly and without too much deliberation between team members, which in turn allowed us to make good progress. This boosted team morale and improved our attitude toward the project in general, and we hope to carry this trend into the next sprint.
- Communication among team members
    - We corresponded with each other mostly via Slack and were able to stay up-to-date with work done and with problems faced. Then, when we came to meetings, we could discuss our correspondence outside of meeting times effectively instead of waiting and explaining all our problems in person. This helped our meeting efficiency toward the end of the sprint.

What didn't work so well this sprint AND WHY?

- Technical issues
    - Kenzie had lots of trouble getting everything set up. Rails does not work very well on Windows (a demonstratively plebeian OS), so she and Haylie had to install Ubuntu on a virtual machine. Learning how to use Ubuntu and install the required technologies was difficult. Kenzie accidentally skipped setting her password when installing MySQL, which made it impossible to set up a server. After uninstalling and reinstalling MySQL, it didn't prompt her for the password again. We were not able to figure out how to reset the password, so we ended up reinstalling Ubuntu entirely. Once Haylie and Kenzie got everything set up, they were behind in knowledge of Rails.
- Long daily scrums
    - Initially we were unsure exactly how a daily sprint should go. We ended up going into great detail about each person's progress, and especially what was getting in their way. Instead of letting them say it and moving on, we often tried to fix the problem right then, which made our meetings exceptionally long. Toward the end of the sprint we increased efficiency and the daily scrums were better.
- Gap in technical knowledge between team members
    - This was caused by the technical setup Haylie and Kenzie had to do. Once everything got set up, development started going more smoothly.

What the team plans to <u>change</u> 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 plan to do more pair programming this sprint than we did in the last one; now that everyone has all their technical problems figured out, we can effectively program in pairs/groups. This will help because then we can all have high velocity instead of just one or two of us, and we will all understand RoR better because we will all be working on it together.
2. Sticking to a more strict time box and agenda as proposed by the Scrum Master will help our meetings stay short and concise. We tend to say we're sticking to a constraint but we sometimes go over it, which is unnecessary and leads to less work being done.