Team Name: The Procrastinators

1. Planning Poker did help out with story point estimation. There were no impasses on point values. The highest difference between 2+ members on values were around 5-7 points. The story this happened for was “Perform log operations in arbitrary bases.”
2. Most of the meetings took place in person, with occasional people missing due to conflicting schedules, those people were either filled in over groupMe or at the next scrum.Most of the meetings took place in the evenings with the scrum starting between 5 and 6. The scrums usually lasted between 15 and 30 minutes followed by some coding, sessions usually lasted less than 2 hours in total.
3. How was using test-driven development? Did you find that your code was of better quality? Would you use TDD again or do you prefer writing the code first? Was it hard to write test for some stories/acceptance criteria?
   * It is not as easy as just writing code. It makes more sense to have something to test than have a test for something that doesn’t exist yet.
   * The code would have been no better if it were done without it, but it probably would have been faster to not use TDD.
   * We would prefer not to use TDD again if given the choice.
   * It is difficult to write any test without pre-existing code to test.
4. What was your velocity for your sprint? How many story points did you plan to accomplish and how many did you actually accomplish? Did you finish everything for your sprint? Is your product potentially shippable? Did you get all the core/most important stories accomplished? If you were to do another sprint, would you increase or decrease the amount of story points in your planned workload for that sprint (i.e. would you attempt to increase or decrease your velocity)?
   * The velocity of this sprint was 66. We planned to accomplish 79 story points, but we only accomplished 66 of them.
   * No we did not complete everything for the sprint, one task was never started and another never made it through testing. Our product is potentially shippable.
   * We did get the core and most important stories for the project accomplished.
   * We would decrease the planned story points, and try to be more accurate with the story points assigned. Many of the tasks did not require as much work as originally estimated, while some required more.
5. Estimate the average cycle time for your stories. This doesn't have to be 100% accurate but if you start keeping track of that data now, you can be more accurate when estimating at the end of your sprint. Also estimate the average lead time for your stories. For only doing a single sprint, cycle time would be more helpful for process improvement and I would imagine lead time would be more helpful over the course of several sprints and overall over the course of the project, but I want you to get some exposure calculating both.
   * Our average cycle time was about 1.437 days.
   * Our average lead time was about 4.1875 days.
6. Using the metrics of velocity and average cycle time, could you improve on your implementation of the scrum software process? What kinds of things could you do to increase your velocity and decrease your cycle times? Would you follow Scrum again or would you make some changes and perhaps introduce some concepts from other methodologies? Could concepts from Kanban, Scrumban or XP help? Explain all your answers going into sufficient detail so I can clearly see you understand these methodologies and their pros/cons
   * Using JIRA to manage our tasks as we worked on them could help with the scrum process. To increase velocity or decrease cycle times more time could be spent working on the project.
   * Concepts from scrumban could help if we had a static workspace with a board that wouldn’t change. It would help everyone visualize the workflow and having WIP limits would help in getting tasks through testing.
7. We did not receive JIRA accounts until the last 2 days of our sprint, so we did not use it. We do not have any of the charts or reports from JIRA.