<3

Heartrate Behavior and

Analysis Tool (HBAT)

Sprint 2 Retrospective

Ruhana Azam, Manoj Polisetti, Rajith Weerasinghe, Phillip Thain

Team 3

# What went well?

This Sprint was successful on multiple fronts; the project is now organized with Maven, allowing for other users to use our open source software. It is also more robust, and produces understandable output. These successes are highlighted through the user stories below:

**As a researcher, I would like to process heartbeat data in order to analyze it against behavioral data.**

* For this sprint, we fixed the issues we had with the sync time and also figured out the answers to most questions we had about the product from Professor Tonnsen. This helped us to fix things in our code from the previous sprint to make it fully functional.
* This sprint, we spent time cleaning up the algorithm code. All bugs from the previous algorithm code have been fixed thus it is now able to calculate phases. In addition, we have met with Professor Tonsen and cleared up any doubts about the algorithm and have implemented the changes.

**As a researcher, I want to see the output file in spreadsheet form from within the GUI window.**

* The output page successfully displays the output data in a table.

**As a researcher, I would like to know if there are any errors with my input.**

* During this sprint, the user inputs would be validated before the data gets processed. The user is shown an error and then has the opportunity to correct it without the program crashing. For example, the files specified must exist before moving on.
* The algorithm code will throw errors if there are any situations in the data that could not happen in real life or algorithm could not calculate. In less serious cases where the algorithm is unable to calculate something, the code will output a -1.

**As a developer, I would like to be able to create a simple way to build the project and test it all it once.**

* This user story was completed by converting the project to the Maven framework. With Maven, making changes to the project, then testing it and building it have all been reduced to a couple simple keystrokes. This is useful for current development, as well as future developers, as the software will be open source.

**As a developer, I would like to easily be able to create test cases and verify that the software is functioning correctly .**

* We worked on individual test cases for the MainParser.java and the Algorithm.java functions to make sure developers have something to test for in their version of the project. This has helped us create insure the algorithm is reliable and also robust in a variety of situations.
* By moving the project to the Maven framework, writing Junit tests are now a cinch. We simply write them according to the Junit standard, and then they can be run against the project every time the project is built.

**As a developer, I would like to be able to close/transform GUI forms in order to reduce clutter on the desktop and improve performance.**

* Now there is only one or two windows open at a time during the lifetime of the project, reducing unnecessary clutter of the desktop.

**As a researcher, I would like “tool-tips” for features I may not understand.**

* There is an option to toggle tooltips, which enables detailed descriptions when hovering over a textbox.

What did not go well?

Although the work over the past few weeks resulted in improvements of the software from last Sprint’s release, our team did not achieve all of the goals we sought to complete. For example, we underestimated the length of time it would take to convert the project to Maven. It took much longer than we thought it might because we kept running into build and runtime errors that we didn’t fully understand. However, now that it is working, future development will be much more straightforward. Here are a few stories that will be completed in the following weeks, as we did not leave adequate time for them during Sprint two:

**User Story 4:** As a researcher, I would like to be able to create visualizations of the data.

* We were not able to find a good way to communicate with an executable in another language (Python) to create visualizations. And because it was such a low priority for our stakeholders, we chose not to do it this sprint.
* We are currently looking into other options such as Google Charts, which supports Java. However, issues can arise because it must be connected to the internet, which may be in violation of HIPPA.

**User Story 5:** As a researcher, I would like to check on multiple infants’ data in a tabular format.

* We failed to implement this feature because we realised that the way our program is set up, we aren’t currently allowing users to open up multiple files at once anyway. Implementing this user story will require some slight redesign within the GUI to more line up with our original intentions outlined in the design document.

How can we improve?

Compared to last sprint, we were felt far more organized as a result of more regular meetings (We met at least three times a week each week) and better communication. However, there is always room for improvement.

One way we can improve our work in the next sprint is to be more generous with the amount of time we estimate tasks will take. The past two sprints we have given ourselves a little bit more than we actually are capable of completing within the given time constraints. It would be better to give ourselves more realistic goals and focus on completing things that work well, over rushing to finish buggy features.

Our work will be ultimately improved by working together more often. We have had discussions about how we tend to be more productive while working as a team than working individually. Therefore we plan to meet more regularly to work on development throughout the next sprint.