Overall, working as a collaborative team of 4 proved to be a challenging yet rewarding process. As we learned in class pertaining to the principles of Agile Development, people are valued more than processes and tools. The first step to establishing a productive coding environment was getting to know each other’s personal strengths, enthusiasms, and weaknesses. This was an integral part of establishing what each person’s role would be in completing the assignment.

The first process was to create the extended state machine model. This was essentially the first draft of our group mindset regarding how we would tackle converting the example stopwatch to the Android timer application. Once we had a better understanding of event based programming both through our group discussions and class lectures, it was time to actually go in and implement the code.

As stated, the extended state machine model really was a first draft. While it was absolutely necessary in opening up the concepts that we needed to grapple with, it was only the first step to accomplish the task at hand. It soon became a matter of understanding exactly what the test cases were pointing us towards, and how we would get them to pass. Another primary aspect of the project that the extended state machine model did not address was the classes that we needed to import, and how to work with each class (specifically those regarding the Android framework). This was an entire beast in itself.

While creating the model was not even half of the process, we still believe that it is more effective to create it first before coding. This is because creating the model gives us an overall idea of what we are going into— a starting point, if you will. It would be a lot more chaotic to tackle the project head on from an algorithmic perspective, without having an understanding of the project as a whole. This could lead to issues with time management, as well as implementation. Now that we have code and the model, it is only natural that we are able to compare the two and see where we could have improved in our drafted model. There are many details that could have been added to the model that could have helped point us towards the algorithmic next steps required to get the tests to pass. However, this is not the point of creating the model. In the end, we are in agreement that working as a development team brings specific challenging elements, but is more ideal in reducing individual computational labor and creating a solid end product.