As a group we decided upon using Agile (Scrum) as our chosen methodology. After doing a lot of research as a group and discussing what we wanted to do as a project and how we code as a group, we found out this was the correct thing to do. By allowing us to release separate working products each time we had a ‘sprint’, this allowed us to look at how much work we had to do for the next release as defined by our project milestones which were tracked separately. By using this method, we could bugfix on each release and also have pre-defined releases, which gave us a good scope for the project and allowed us to work on priorities for the next release. Also, by using Agile as a general methodology, we could take a step back at various points throughout the lifecycle and change how we were working and what we wanted to change.

One of the drawbacks of using Scrum is the fact that it is constant code development with not much time in-between. As soon as one release was done, another release was being worked on with no much time to evaluate what we did well, what we didn’t do well and what we need to improve, which lead to confusion after releases on what has been achieved and who has contributed effectively. Due to this, a different methodology might’ve been the most appropriate for this kind of activity, one that allowed more reflection and evaluation.

By developing a Gantt chart that mapped out our software development life cycle alongside our chosen methodology we were able to see how timings were working and how far along the cycle we are and where we should be aiming for next. It also allowed us to simulate a real life project and resource map to take a look at how different people will be working on different things. It proved that we were terrifically over subscribed for the workload we were taking on and if this were a real life company there would be a team of at least 10 working on this at any one time (when taking into account working days, weekends, overtime and payment).

Developing a detailed flowchart was a good move as we were able to look at how different elements communicated with each other, in the same way we used the use cases. The flowchart accurately matches the final version product which is a huge benefit as it shows we had a structured development process for our project and also shows our chosen methodology, although being impractical at times, worked. The flowchart could’ve included slightly more detail to allow alternative flows, as set out in the use cases in appendices.

Deadlines and milestones were predefined at a very early point in the life cycle to give us an idea of how far we were getting, how much work we had left to do and where we were falling short. This allowed us to chase up issues that were coming up ASAP so that we didn’t fall behind on the deadlines and by having separate releases, inline with our methodology, allowed bug fixes and evaluation. Despite the positives, it was difficult to keep on track of the milestones at certain points and when members were falling behind on documentation work.

Github turned out to be the fifth group member during this process. Although there was issues with branches early on in the process (especially when working on the same file which really isn’t easy), it allowed remote working and helped us to track the progress the group was making. Using Github instead of Dropbox was a lot better as this ensured that no one could affect the code itself and then it being lost forever as Github allows you to go back to a previous commit that basically backed up each and every implementation made.