**2.4 Development Strategy.**

A clearly defined development strategy is key when partaking in the development of an application. It defines the rules, helps to monitor time and ensures some task are not overlooked. In technological development there are two main strategies that are followed. There are the Waterfall model and the Agile model. This section aims to provide a background on each model and draw a conclusion as why the selected development model/strategy was used within this project.

**2.4.1 Waterfall model.**

The waterfall model, which comprises of linear successive phases (requirements engineering, design, implementation and testing) with each phase needing to be complete before the next phase can begin. It provides advantages such as providing a clear set of requirements upfront that allow milestones to be easily monitored (Sommerville, 2011). Furthermore, it provides a clear audit trail with responsibilities being easily understood by all involved and can be used to easily define associated costs.

**2.4.2 Agile model.**

The agile model is software development approach based on iterative development. Agile breaks tasks into iterations or sprints that typically last for two to four weeks, with a situation review at the end of the cycle, (van Vliet, 2008), *figure 6*. Each iteration involves working through a full software development cycle including planning requirements analysis, design, coding, and testing. The main aim of agile to reduce the amount of upfront planning and documentation and via each iteration it builds upon the functionality created in the previous sprint. This allows for continuous delivery of useful software and functionality can be changed depending upon changing circumstances.

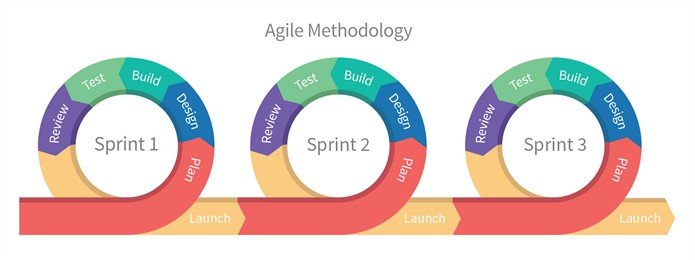


Figure 6 – Agile Model

**2.4.3 Selected Development model.**

Due to the timeframe and size of the project, the Agile model was deemed more suitable for the project as it focuses on individuals and interactions over processes and tools and working software over comprehensive documentation. The planning stage associated with the Waterfall model is usually a long drawn out phase at the beginning of the project that provides requirements, specifications and documentation. While this can be useful in large scale projects, it does require a lot of work at the beginning and it does not allow for changes throughout the process. This process can be useful in large scale projects, but it should be remembered that this is an individual project and changes to requirements and functionality need to be embraced quickly. Furthermore, at the beginning of the project, the developer was not entirely familiar with the chosen technology making the planning stage, related with the Waterfall model, slightly more difficult. Instead, by using the Agile model, changes to design and functionally can be quickly implemented in each iteration as knowledge is improved.

**2.4.4 Agile Development within the project.**

As stated before, the Agile development takes the form of small incremental cycles and throughout this project these sprints took place. Within this project 5 sprints took place each lasting two to four weeks and with supervisory meetings taking place within this time frame it provided the perfect opportunity to analyse the current process of the project and change requirements if required.

At the beginning of the first sprint a product backlog was created to act as a list of the work that needed to be done to the application. In each sprint a smaller list, or sprint backlog, was taken the product backlog and those features and requirements were implemented. The product backlog consisted of user stories and these were directly implemented as tasks for the project.

A burnout chart was used a visual representation of tasks to complete in the product backlog vs time within the project, see *figure 7.* At the beginning of the project it can be seen that a large number of tasks were not completed within the first sprints. This is due to the aforementioned lack of experience in using the selected frameworks, however, as familiarity increased it can be seen that the rate of task completion sped up.

Figure 7 - Burndown chart showing percentage of tasks needing competition vs ideal and actual

As each sprint follows contains each element of the waterfall model e.g. implementing and testing, it allowed for continuous development and testing of the existing software. As more tasks were completed in later sprints it allowed for more vigorous testing including alpha testing that allowed any bugs not recognised by the developer to be fixed. By carrying out testing at each stage it meant that any potential defects could be quickly identified and take less time to fix (van Vliet, 2008). As well as this, any input from testers regarding functionality could also be implemented.

After each sprint was completed a sprint review was carried out. A sprint review looks at the pervious sprint, validating the system and decides what user stories should be implemented in the next sprint. By carrying out the Agile development strategy throughout the project it meant that all user stories were implemented to high standard, removing any potential defects at the outset.