Software Lifecycle Model

Waterfall Model

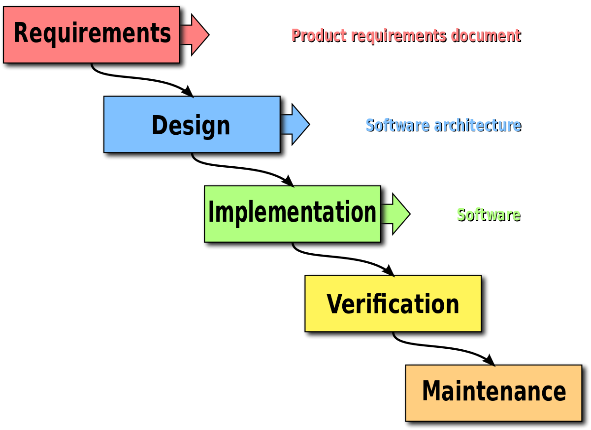
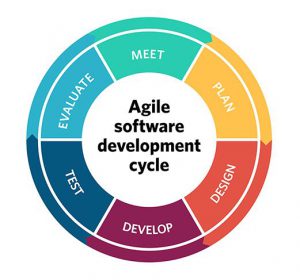
 One of the first software lifecycle models we researched at was the Waterfall model. The model breaks down project activities into linear sequential phases. Each of the phases is dependent on the previous phase and corresponds to a specialisation of tasks. It uses a clear structure when compared with other methodologies. A team must complete an entire step before moving onto the next. If any problems are encountered, they are addressed right away because it stops the progression in its tracks, leaving the team with a complete and polished product as a result. Another advantage we found was that the team can start right away without any steep learning curve to slow the process down. The model determines the end goals early and transfers information well as it is a highly methodical approach.

Figure 1 The Waterfall Model

When investigating further into the Waterfall Model we found that any changes that occur will be difficult to implement to our design, leaving no room for unexpected changes or revisions. We also found that it delays testing until after completion of the project. For both reasons we have decided to look further into other approaches we can possibly use.

Agile Approach

The next software lifecycle we decided to research, and later decided on was the Agile Approach. We found that Agile methods generally promote a disciplined project management approach that encourages frequent inspection and adaptation. It adapts a leadership philosophy that encourages teamwork, organisation on an individual level and individual accountability. It seems the best option to produce a high-quality software with a rapid delivery.

Agile development refers to the development processes that correspond with the Agile Manifesto. The Agile Manifesto has a number of principles some of which include; Delivering working software frequently, trusting other team members to get their tasks done involving frequent communication amongst the team and lastly welcoming changing requirements even at late stages in the development. The short timeframe for development suits this project along with the fact that we are all equal members of the team in terms of workload. There may be aspects of the project that need to change later in development such as extra implementations that get introduced at a later stage in development.

Figure 2 Agile Process