**Waterfall Lifecycle**

When further investigating the Waterfall model, we noticed a few flaws in the idea which wouldn’t be appropriate for our project. They included:

* Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage of the project.
* Not a good model for complex and object-oriented projects.
* Not suitable for the projects where requirements are at a moderate to high risk of changing.
* Poor model for long and ongoing projects



**V-Model Lifecycle**

We also decided to take a look into the V-Model as well but due to the following problems listed below, we decided not to take this method either.

* Very rigid and least flexible.
* Software is developed during the implementation phase, so no early prototypes of the software are produced.
* If any changes happen midway, then the test documents along with requirement documents has to be updated.

