Ngoh Man Ling

Software Development Life Cycle

Software development life cycle is a process which includes planning, analysis, designing, building, testing and deployment. This process are often used for a software project. First, we will need to set the requirements for the system. We can use customer inputs and market surveys to come up with the relevant requirements needed for the software. Besides that, we will also need to consider the quality and risks when creating the software.

Second, to analyse and to document the required requirements in a document. This will include detailed description of the data needed, input and output. This two phrase are important as poor analysis and specification of the requirements will affect phrases in the later stage. For instance, they may forget to put some of the requirements needed but they had already reach the testing stage. Some of the models will not allow for changes in the previous stage which will cause them rework on the whole process again.

Third, before we implement the system, we will first need to design the system in order to build a robust system. The design of the software will includes architectural modules, data flow of the system and the interfaces. By designing, it will give us a clear idea how our system will look like which makes it easier for us to implement.

Fourth, to build up the system using programming languages such as C++, C#. To add on, a well-written code will help to reduce the error occur in the testing and maintenance phrase.

Fifth, to test the product before deploying to the market. There are two test which are program and system test. All of the program must be tested individually with the prepared test data first. Error must also be debugged in the program testing if there are any. After finishing with the program testing, system testing will then be executed with the actual data. The output of the system will be analysed and fixed if there are any errors. This will help to ensure the software reliability and quality.

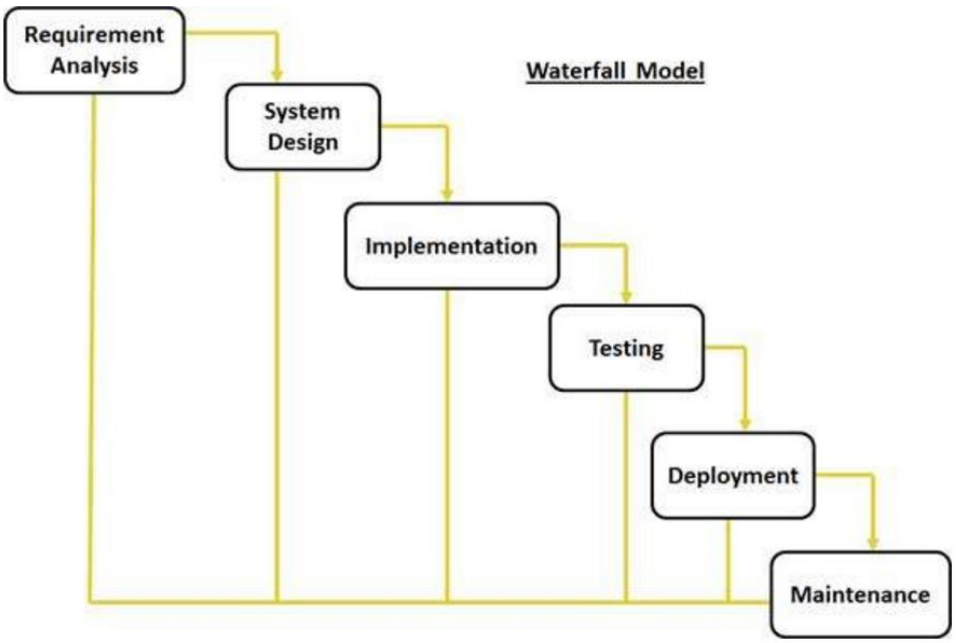
Lastly, to deploy the system to the market. Business may first choose to release the system before the official date of release. This will allow improvements to be made before the actual release. Maintenance of the system after deploying are also important for good user’s experience. For example, to fix bugs and update system version. There are many types of software development model we can use. Below are some of the model that are used.

Waterfall Model

The waterfall model will help to give you a view of SDLC in a linear sequential flow. Each phrase of the model need to be completed before going on to the next phrase. After completing each phrase, we cannot go back to the previous phrase to make changes due to the rigidity of the model. Besides that, waterfall model are use when the project is short and requirements of software are clear.

The benefits of using this model is because it is simple which makes it easy for us to understand. It has clearly defined stages and makes it easy for us to assign tasks. This model help us to save time and resources as well. For example, I am creating a restaurant POS system for Delonix Regia Hotel. There will be less chance that I will need to rework on the system as development is processed in sequential manner.

However, processing in sequential manner makes it difficult for user to go back to the previous phrases if they want to change the requirements. They will need to restart to the beginning of the phrase again which will waste time. Hence, the requirements set at the start of the process must be clear and detailed. Moreover, these types of model are not suitable for long project because software will only be produced during the end of the cycle. They may face many errors which requires more time to fix.

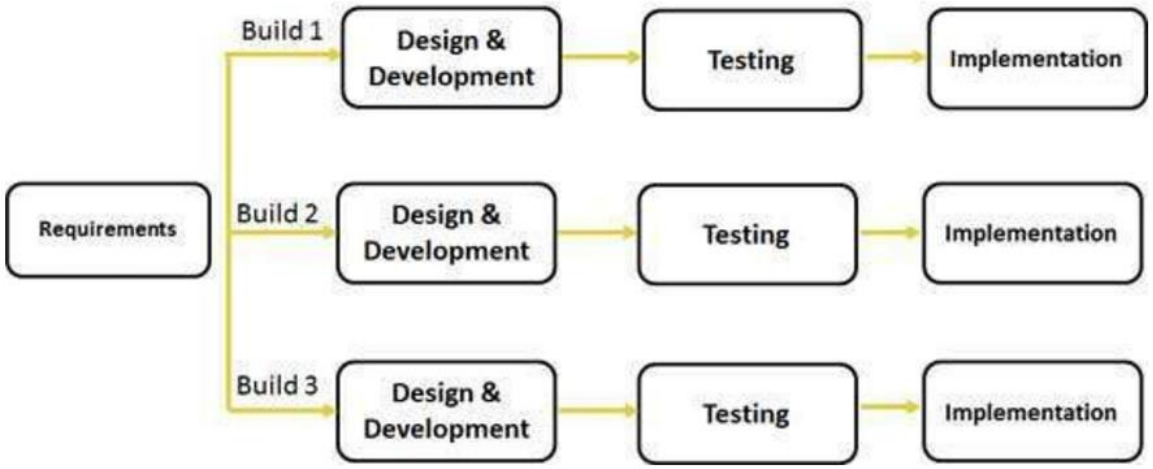


Iterative model

Iterative model process will first start by stating the requirements and implementing part of the software for further enhancement or requirements of the software. The software will then be improve with the required features and a new version of the software will then be released again. These process will then be repeated again and again until the complete system are functional and robust. This model are different from waterfall model where phrases are only gone through once. For each iteration, the module will goes through the system requirements, analysis, design, implementation and testing. This model are use when working on new technology and resources are not available.

The advantages of this model is that we will be able to get back the results of the software early. The issues identified in each iteration can be solve and released the new version of the software. This allows us to constantly improve on the software until it is working fine.

However, we may be face with architecture issues as the full requirements are not specified at the start of the life cycle. These model will not be suitable for projects that need clear defined requirements at the start of the phrase. For example, the project that we are working will not be suitable for this model as we are creating four sub-system in the software. We will need to gather the full requirements of the software to save times and resources.

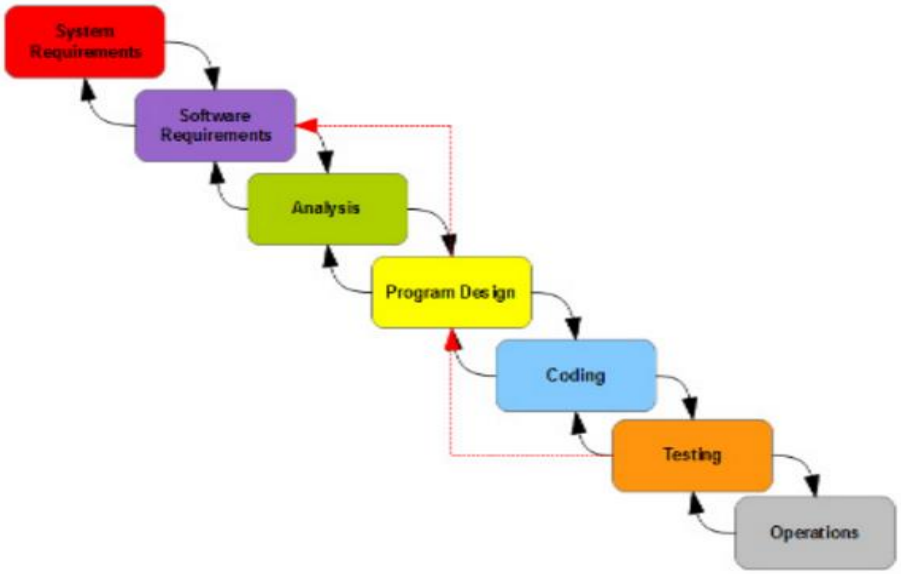


Agile model

Agile model uses an incremental approach. They first start with a simple design and work on the modules which are done on a weekly or monthly basis. These module or phrase are called sprints. At the end of each sprint, project are evaluated and tested. This helps us to identify bugs or issues in the software. Customer feedback are also use to enhance the system also before running the next sprint.

The benefits of agile methodology is that it allow for changes during the process. We will be able to add features easily if the client requested it. This model are even better for creating sub-system of the hotel management system than the waterfall methodology. With the flexibility of making changes at the previous phrase, it allow us to save time on improving and fix issues of the hotel systems.

However, we may need to refactor the system frequently as the initial architecture design are not fully specified. To add on, it also requires communication with client most of the time. If the client are not sure on their requirements, system developer can be going in the wrong direction.



References

Waterfall vs. Agile: Which is the right development methodology for your project? (n.d.). Retrieved May 09, 2016, from http://www.seguetech.com/blog/2013/07/05/waterfall-vs-agile-right-development-methodology

Agile software development methodologies and how to apply them. (n.d.). Retrieved May 09, 2016, from http://www.codeproject.com/Articles/604417/Agile-software-development-methodologies-and-how-t

SDLC Overview. (n.d.). Retrieved May 09, 2016, from http://www.tutorialspoint.com/sdlc/sdlc_overview.htm