Chang Won Kit (1) 1703879E C17B05

What is SDLC?

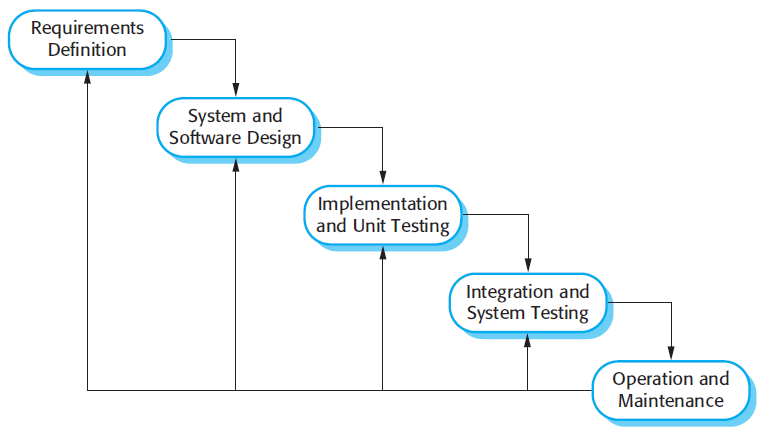
SDLC is a scheme that provides guidance on the tasks needed to be fulfilled to create quality software. It is a structure that breaks down the steps to maintain, develop and improve software. This defines the life cycle of a software and the various processes that occur within development.

3 types of models and how they work. Short summarized descriptions.

Waterfall model

The waterfall model is a rigid linear process where each step of the development process is divided into separate phases.

According to the principles, processes are finished before the next are started. There are several processes in the Waterfall model. (Requirement Definition, System and software design, Implementation and Unit testing, Integration and system testing, Operation and maintenance)



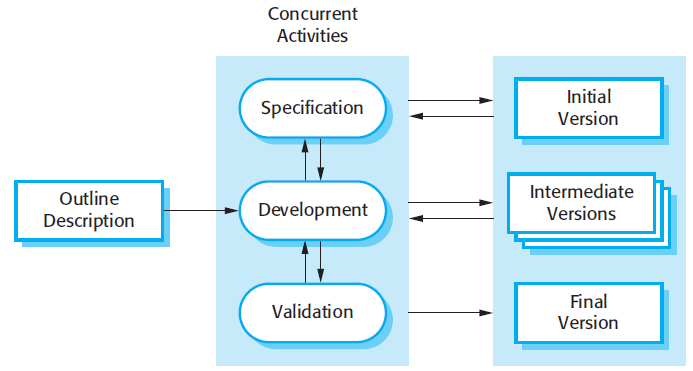
There are several disadvantages to this method, during the processes, phases may overlap. For example, during design, problems with requirements can be identified, and during coding, some of the design problems can be found. Therefore, if a linear process is observed, continuous changes will have to be made to documents to enact the changes made, after each process. Also, if an error was made in the early stages of development, it will be a lot harder to amend changes as more processes are completed.

The waterfall model is advised to be used only when the specifications and processes of a project is clearly understood and is not subjected to major changes. The rigid process model makes it hard to accommodate changes.

Incremental Development

Based on the idea of developing an initial prototype, while letting user input influence the development of the application, until the final version.

The processes are not separate but combined and influenced by user feedback in all the processes. Each stage of increment completes certain functionalities. Early increments develop the fundamental functionalities of a software. As increment stages progress, more functions are incorporated into the software.



Customer input allows the evaluation of the software at early stages, this can reduce the chances of wrong specifications being carried over and affecting future versions of the software. This can be increasingly harder to amend as the software is being built around these specifications.

Reuse-oriented Software Engineering

It attempts to reuse existing designs or code that has similar requirements of the software. Modification and integration take place to integrate software into the system.

Requirement specifications and validation phases are similar to other software processes; however, the intermediate phases are different from other models.

There is component analysis, which searches for components that match the specifications. Requirements modification, which would involve user input, as the components available will have to complement with requirements. System design with reuse, the framework of the system is designed and modified to complement the software Development and integration, the software, components and system are integrated to produce a new system.

https://medium.com/omarelgabrys-blog/software-engineering-software-process-and-software-process-models-part-2-4a9d06213fdc