**Chua Cheng Yu**

# System Development Life Cycle (SDLC) Assignment

The **systems development life cycle (SDLC)**, is a term used in systems engineering and software engineering to describe a process for planning, creating, testing, and deploying an information system. The systems development life-cycle concept applies to a range of hardware and software configurations, as a system can be composed of hardware only, software only, or a combination of both. SDLC aims to produce quality systems that meet or exceed customer expectations, based on customer requirements, by delivering systems which move through each clearly phase, within scheduled time and cost estimated. Therefore, the system will be fast and efficient to deliver all these to the customers.

The first process is the requirements engineering. This process is to find out what the user, which is either the clients or the customers wants the software supposed to do it.

The second process is the analysis. The goal is to determine where the problem is so that it could fix the system. This step involves breaking down the system in different pieces to analyse the situation, analysing the goals and analysis where the problem come from. Breaking down what needs to be created and attempting to engage users so that requirements can be defined. It also use a combination of text and diagram form to depict the requirements.

The third process is the design. It produce a representation of an entity that will later be built. It also include architectural design, user interface design and the database design.

The third process is the coding implementation. This process provides detailed designs which converted into instructions written in the programming language.

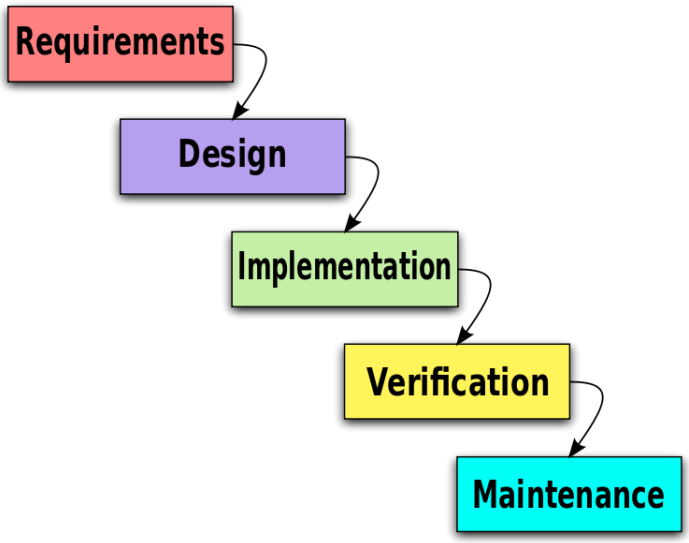
The fourth process is the testing phase. It is to ensure the software is reliable after testing and to also ensure the software meets the user’s needs.

The fifth and last process is the deployment phase. It is the final application that is distributed among a group of selected clients or customers prior to the official release date.

There is also one more process is the maintenance phase. This maintenance phase consisted of two parts, remedial maintenance and the adaptive maintenance. Remedial maintenance help to correct the errors in the main system. It needs the maintenance is also due to inadequate testing of the system. Adaptive maintenance can adapt applications to change in the environment for example like the computer or the operating system. It also can improve, change or adding new features in it. It is also believe that the software is somehow “soft”, hence easy to modify.

Furthermore, there is also process model in the SDLC. The process model is like a plan of what steps to take to develop a project. We can also use the models to predict what will be done. It can also use it to analyse the current development process and make adjustments to improve on the system. There are a few process models. The models are waterfall, prototyping, unified process and agile methods. Out of all this models, i chose the waterfall, prototype and agile method models.

**Waterfall**



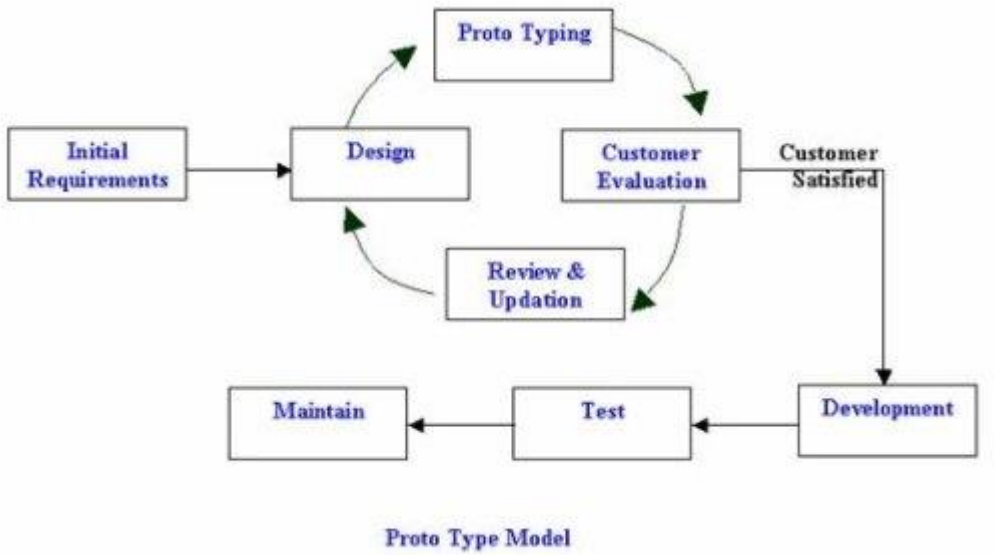

There are also pros and cons on this waterfall model.

**Pros:** It can divide complex task into smaller and hence more manageable tasks to maintain control. It also can make each task to produce a well-defined document. It is easy to control and monitor because we deal it with one activity at

a time. Getting the requirements and design out of the way also improves quality and it is much easier to catch and correct possible mistakes at the design stage than at the testing stage, after all the components have been integrated and tracking down specific errors is more complex.

**Cons:** We can only see the product only at the end, which makes it no opportunity to validate the user requirements at the early stages of the development. Also, if a problem is discovered at an early stage, nothing can be done about it like to modified or update it. It also does not stress the need for any anticipating changes. Customers does not really know what they want up front, rather, what they want emerges out of repeated two-way interactions over the course of the project.

## Prototyping




Proto Type Model

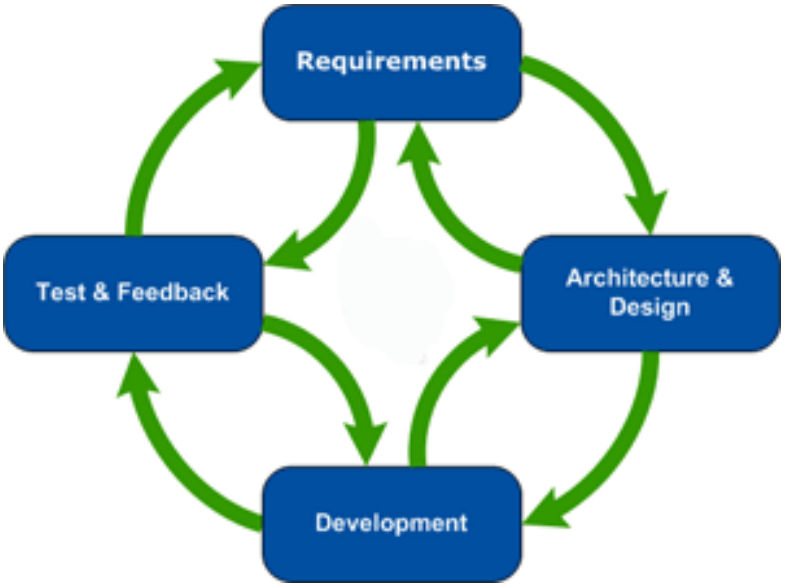
A prototype is a scaled down initial version of the target system. The goal of this is to clarify its requirements. It can be also use to design the user interface, demonstrate its feasibility, verify that the new technology will work perfectly.

**Pros:** It can clarify the user requirements and also the specifications can be developed incrementally, giving the customers or the client to change their minds. Also, developing the software in the prototype model is that this model allows a high user interface of the customer with the developed system.

**Cons:** The estimating, planning and managing a prototype project can be quite

difficult as there is no regular deliverable. Its continual changes tend to corrupt the software structure and the changes will be become more costly and difficult.
There is two type of prototype. One is throwaway prototype and another is the evolutionary prototype.
Throwaway prototype implementing those poorly understood requirements to build it. After showing and validating with client, we will write a full specification and throw away the prototype. A full scale system is then built based on those specification.
Evolutionary prototyping is to develop an initial system implementing parts of the system that are well understood. It show the developed system to user for comments. Also to refine the system and repeat the process to implement more parts of the system until the full system is completed.

**Agile Methods**




The agile development method is based on iterative and incremental development. In software development, the agile model does not build an entire system at once, but rather develops incrementally.
**Pros:** Encourages active involvement and interaction from key project stakeholders which allows for product build based on priority and accuracy. Continuing the show and tell sessions allow for product agreement from customers and clients. Constant team involvement and structured communication

channels allow for the discovery of progress, changes and collaboration.

**Cons:** In terms of the business, time and efforts will be continually required from product resources. This is crucial to cycle planning and success. Team members must be highly skilled or cross skilled in completing those high level projects as core teams are small. They must also be up-skilled on the agile framework chosen.