Chloven Tan

System Development life cycle (SDLC) is the process of several steps in an information system. The concept of SDLC applies to a range of hardware and software configurations. It is composed of a number of work phases which are used by system engineers and system developers to deliver information systems.

The process that goes on from the start of project to deploy the software system. And the processes are Requirements engineering, Implementation, Testing and Deployment. These are the different phases of a Software Development Life Cycle.

Requirements Engineering will find out what the client or customer requirements for the software and what it supposed to do.

Analysis uses a combination of texts and diagram forms to show the requirements stated from the client and customer. Design is to produce a representation of an entity that will later be built.

Implementation is to produce detailed designs converted into instructions written in the programming language.

Testing is to test the software for ensuring that that it is reliable and will meet the client or customers' needs.

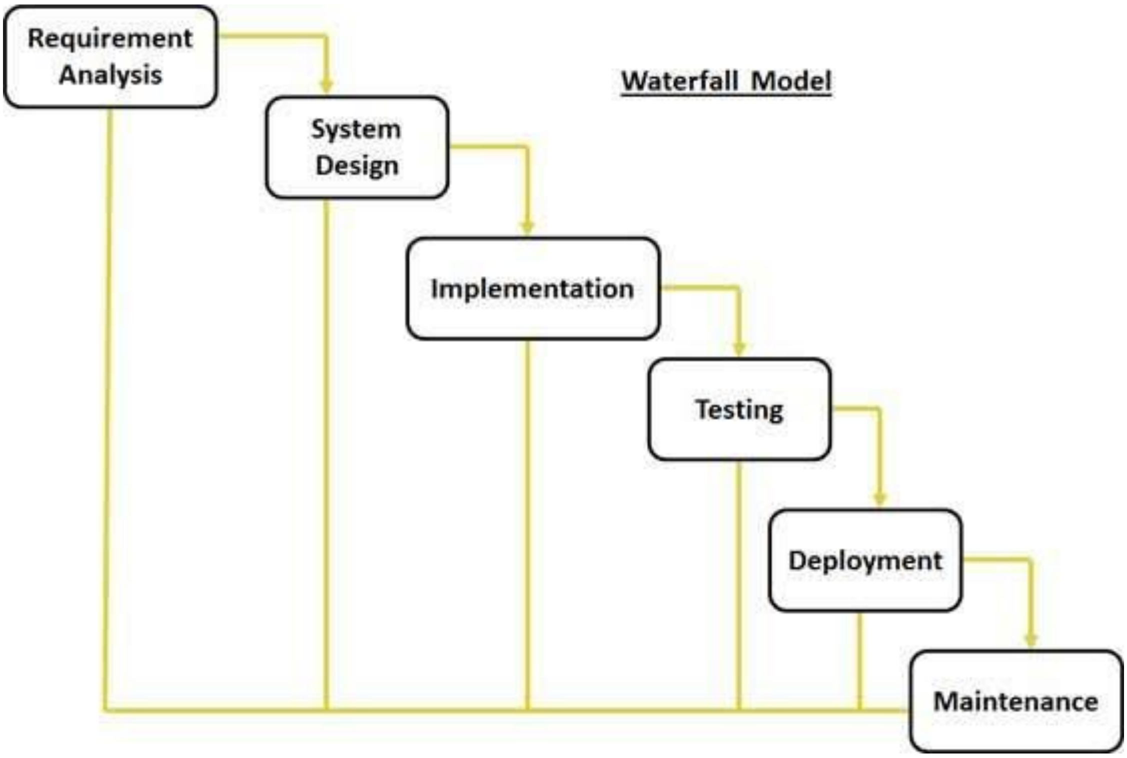
Deployment is when the software application is distributed to a group of selected customer before the official release and is delivered to customer which may involve training as well.

There is one more step which is maintenance and there are 2 types of maintenance. The first is Remedial maintenance which includes correcting the errors in the system due to inadequate testing. Another type is Adaptive maintenance which adapting the application to the changes in the environment like computer, operating system or language. Furthermore, improving, changing and adding new features to the system.

For the Software Development Life Cycle, there are several models called process model which is a plan of what steps to take to develop a project. We can use it to predict what will be done and also to analyse our current development process and to make adjustments or improvements to the software. Process models covered the following: Waterfall model, Prototyping, Unified process and agile methods.

Waterfall Model

The Waterfall Model illustrates the software development process in a linear sequential flow which means that any phase in the development process begins only if the previous phase is complete. The phases do not overlap with each other.



The **advantage** of this model is that it allows us to divide complex task into smaller parts which will make it more manageable.

A schedule can be set with deadlines for each stage of the development and a software can proceed through the process model phases one by one which is easy to control and monitor as we deal with one activity at a time.

It moves from concept through the different phases of the SDLC and ends at maintenance. Each phase of the development proceeds in order. Also, each task can produce a well detailed deliverable.

The **disadvantage** of this model is that it only can see the product at the end and there is no opportunity to validate user requirements at early stages of the development. It does not allow much reflection or revision. It is very difficult to go back and change something once it is in the testing stage.

Prototyping

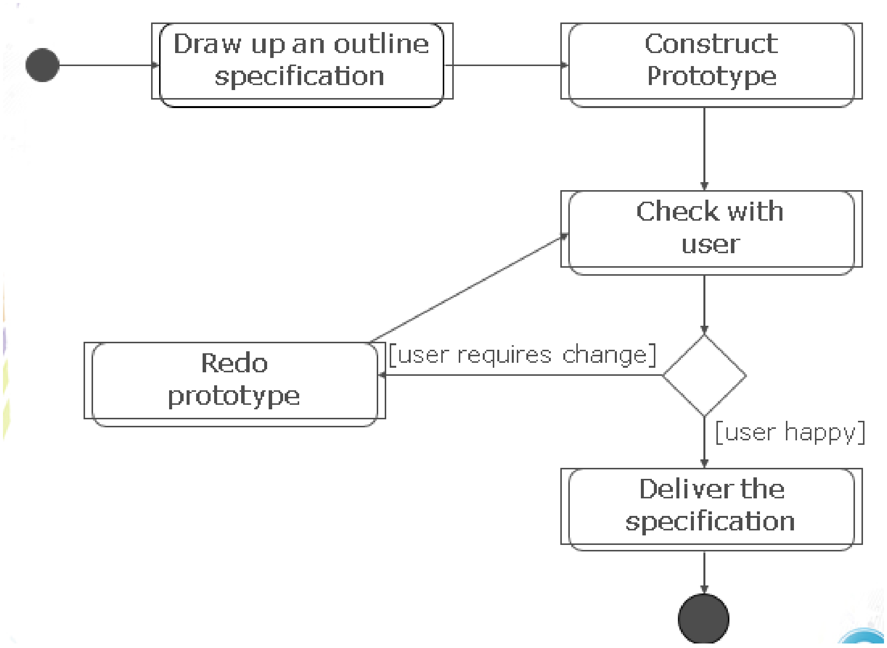
Prototype is a working model of a software and it does not hold the exact logic used in the actual software application. It is used to allow the users evaluate developer proposals and try them out before implementation.

A prototype is a scaled down initial version of the target system and the goal is usually clarifying requirements. It can also be used to design the user interface, demonstrate feasibility, and verify that the new technology will work.

There are different types of software prototype which includes Throwaway Prototyping and Evolutionary Prototyping.

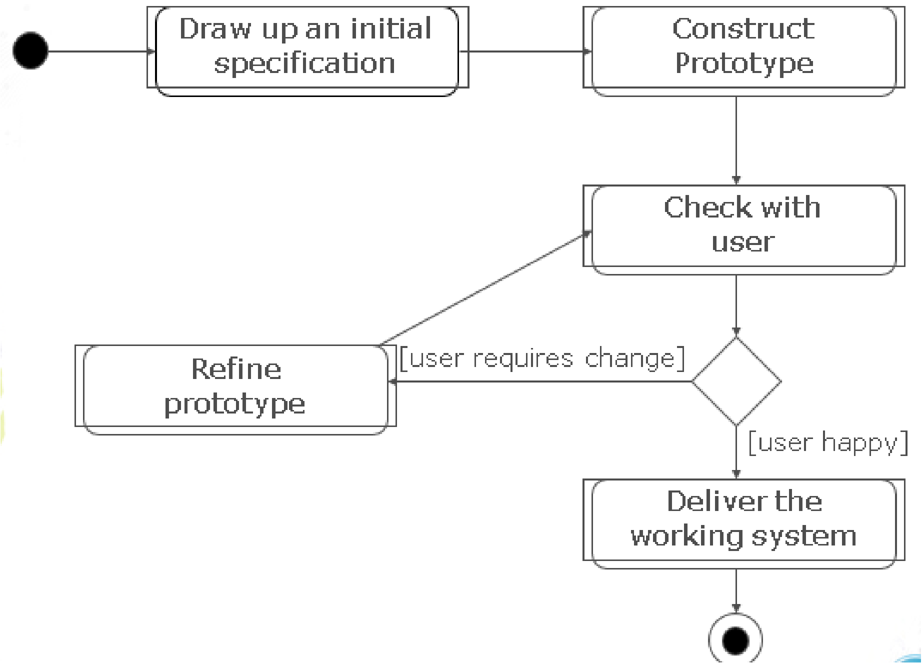
Throwaway Prototyping uses very little efforts with minimum requirement analysis to build a prototype. Once the requirements are understood, the prototype is thrown away and the actual system is developed with a much clear understanding and knowledge of the client or customer requirements after showing and validating with the client or customer which writes a full specification.

The stages of a Throwaway Prototyping:



Evolutionary Prototyping is based on building an actual functional prototype with lesser functionality in the beginning. The reason is that when it is built, it forms the heart of the new system, and the improvements and further requirements will be built. Using evolutionary prototyping only well understood requirements are included in the prototype and the requirements are added as and when they are understood.

Showing the developed system to user for comments to explore requirements for the prototype.



The **advantage** of prototyping is that it increases user involvement in the product before implementation and this will be able to clarify user requirements.

Also, Specifications can be developed incrementally, giving users the opportunity to change their mind while it is in development.

Furthermore, since it is a working model of the system is displayed, the users can get a better understanding of the system being developed.

The **disadvantage** of prototyping is that estimating, planning and managing a prototype project can be difficult as there are no regular deliverables.

Also, there is a risk of insufficient requirement analysis owning to too much dependency on prototype.

Continual changes will tent to corrupt software structure and changes become costlier and difficult.

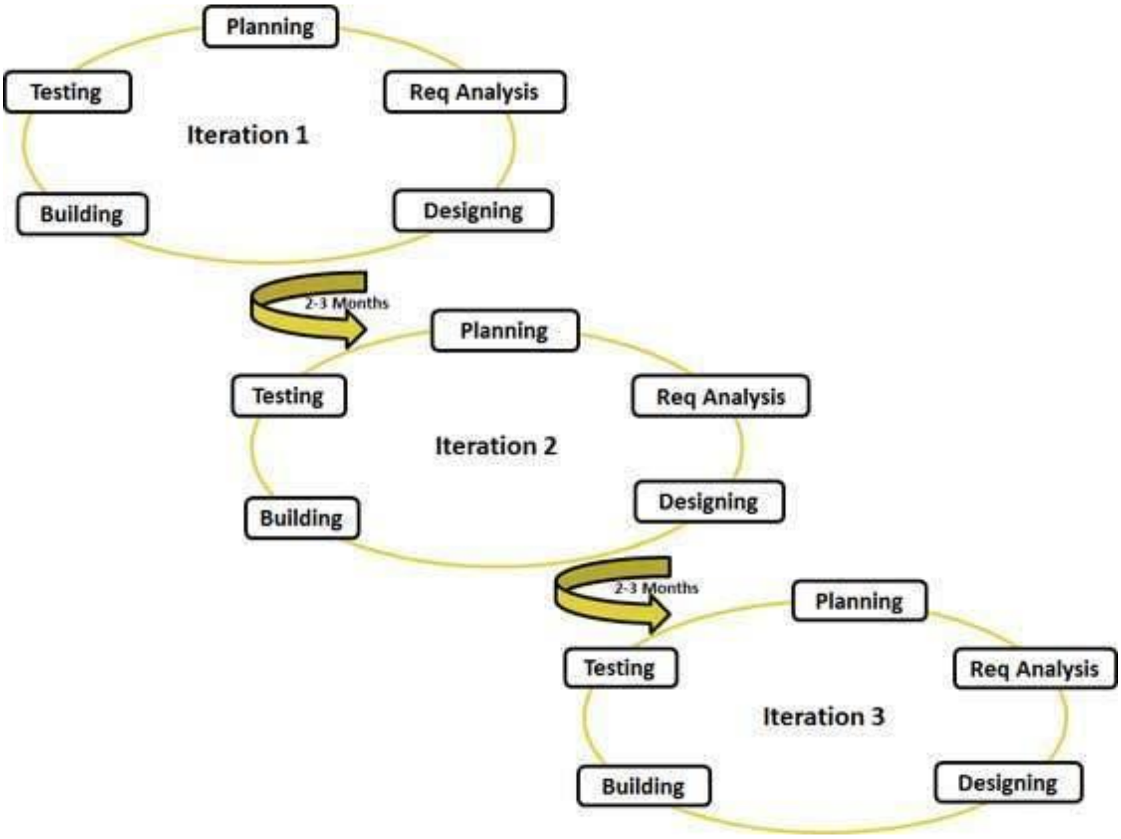
Agile Model

Agile model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product.

It breaks the product into small incremental builds which are provided in iterations. Each iteration lasts from about one to three weeks.

Every project is to be handled differently and the existing methods need to be tailored to best suit the project requirements in agile, the tasks are divided to time boxes to deliver specific features for a release.

Iterative approach is taken and working software build is delivered after each iteration. The final build holds all the features required by the customer.



The **Advantage** of agile model is that the functionality can be developed rapidly and demonstrated. It also has very little minimum resource requirements and is suitable for fixed or changing requirements. Furthermore, it is a very realistic approach to software development.

The **Disadvantage** of agile model is that it is not suitable for handling complex dependencies and it has more risk of sustainability, maintainability and extensibility.

Furthermore, it has strict delivery management dictates the scope, functionality to be delivered, and adjustments to meet the deadlines.