Ch’ng Wai Kit Wesley 5/11/2017

SDLC Individual Writeup

**SDLC (Software Development Life Cycle)**

What is SDLC?

SDLC stands for “Software Development Life Cycle” and it is a process followed during development of a software project. Typically, within a software organization. It is a detailed plan consisting of five phases, namely Requirements engineering, Analysis & Design, Implementation, Testing, and Deployment.

* Requirements Engineering is to find out what the client or customer wants the software to do and achieve. It must be specific and cover every part of the software.
* Analysis is to analyze and use a combination of text and diagrammatic forms to depict requirements specifically. Design is to produce a representation of an entity that will be developed and built on, such as a form of layout. This includes architectural design, user interface design, and database design for example.
* Implementation refers to detailed designs converted into instructions written in the programming language such as Java to make a working software. This means the real code and backend of a software will be written here, along with the designs and interfaces of it.
* Testing is a phase in which the team will test out the software to ensure that it is functioning, reliable, and meet the client’s needs and requirements.
* Deployment refers to the application being distributed among a group of selected customers before release (alpha, beta versions) or other customers. This is the final stage of the SDLC.

**1# Waterfall Model**

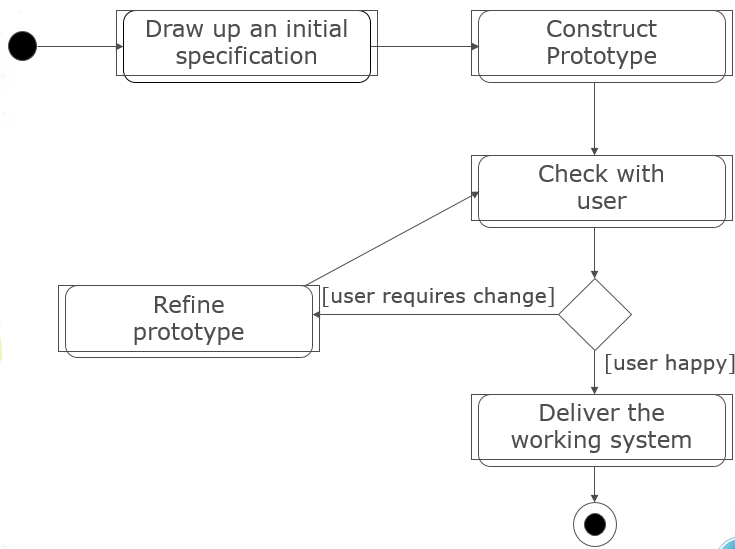
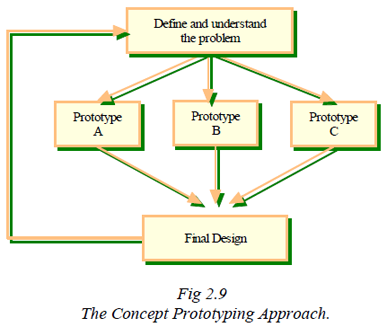


The first software development model is the traditional waterfall approach, which is a sequential design process. It is when progress is seen as a waterfall, flowing steadily downwards through different phases. The phases being - Requirements, Analysis/ Design, Coding, Testing, and Development.

This model can be used to divide complex tasks into smaller and hence more manageable tasks since they are being broken down. In addition, it helps the team identify the deliverables for each task and is easy to control and monitor as it deals with one activity at a time.

However, one can only see the product at the end for the client as there is no opportunity to validate user requirements at the early stage of development. Also, a problem might not be able to be solved if it is discovered at an early stage as it can only be done at a later stage. Lastly, it does not stress the need for anticipating changes.

**#2 Prototyping Model**



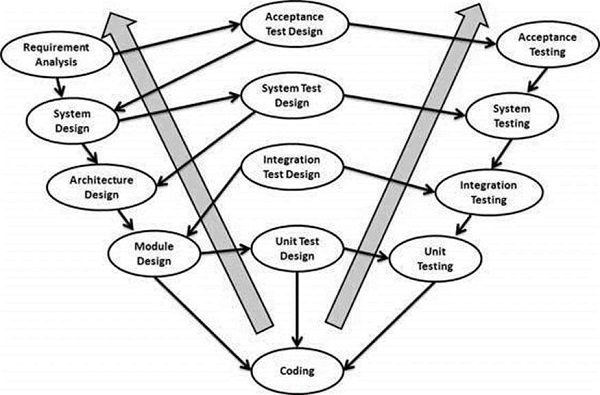
Throwaway Prototyping Evolutionary Prototyping

The second software development model is Prototyping, where instead of developing a full-fledged system immediately, the software engineers start with a prototype which is a simplified version of a software.

The purpose of this model is to clarify the requirements and can be used to design user interfaces, demonstrate feasibility, and verify that the new technology will work for the project.

Prototyping is divided into two types, the Throwaway Prototyping and the Evolutionary Prototyping. Throwaway prototyping is to verify the requirements that are unclear so that both the clients and the software engineers will know exactly what the requirements are. Evolutionary prototyping is to uncover unknown requirements to be built and evolved on and it is especially useful when requirements are hard to specify.

**#3 V–Model**



The third and final software development model is the V-Model. The V-Model is a SDLC model where the execution of processes happens in a sequential manner in a V-shape. It is also known as the Verification and Validation model.

The V-Model is seen as an extension of the Waterfall Model and is based on the association of testing phase for each corresponding development stage. Meaning, for every phase, there is a directly associated testing phase.

The V-Model’s corresponding testing phase of the development phase is planned in parallel. Therefore, there are Verification phases on one side of the ‘V’ and Validation phases on the other side as can be seen in the diagram above.

Similar to the Waterfall Model, the V-Model has different phases. The phases of Verification is Business Requirement Analysis, System Design, Architectural Design (High Level Design HLD), Module Design (Low Level Design LLD). The phases of Validation is Unit Testing, Integration Testing, System Testing and Acceptance Testing.

The V-Model method is advantageous in a sense that it is very easy to understand and apply. The simplicity of this model also makes it easier for management purposes. The disadvantage is that the model is not flexible to changes and just in case there is a requirement change, which is very common in today’s dynamic world, it becomes very expensive to make the change.

It is also crucial that requirements have to be very clear before the project starts, because it is usually expensive to go back and make changes. This model is used in the medical development field, as it is strictly a disciplined domain.

**References**

<https://www.tutorialspoint.com/sdlc/sdlc_overview.htm>

<https://en.wikipedia.org/wiki/Waterfall_model>

<https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm>

<https://www.tutorialspoint.com/sdlc/sdlc_software_prototyping.htm>

<http://istqbexamcertification.com/what-is-prototype-model-advantages-disadvantages-and-when-to-use-it/>