Ang Wei Xiang

1506522J

**Question 1**

Software development life cycle contains a list of steps that is usually used in the software industry to build a software. The steps include on how to develop, maintain, replace and alter or enhance specific software. The main purpose of this steps help to ensure the process of building software base on customer requirements with minimal problems.

In a standard process of a SDLC the first phase is all about planning and requirement analysis. It is important to know the software requirement, therefore gathering information from the organization and users. The information will then be studied and use it to create the requirement of the software and saved it in a SRS (Software Requirement Specification) documents.

The second phase will be about the design of the software and system. In this phase, there will be many design proposed based on the SRS, all the design will be saved in the DDS (Design Document Specification). The stakeholder will then choose the best design approach from the DDS. The design approach must describe clearly of the architectural design such as components that needs to be developed, how does data flow work in the software, how it communicates with the database and any third-party modules. This phase also allows the testing team to plan their test cases on what to test for the software.

The third phase will be about implementation. In this phase, the team will begin to start the development of the software by using programming code base on the design that the stakeholder had chosen. With the design, it will be much more straightforward to code it out in a more structured mannered. The developers must use the coding guideline that is given by their organization. During the implementation, the codes will be repeatedly being reviewed until it met the requirements.

The fourth phase will be about testing. In this phase, the software will be given to the tester to use. The purpose of doing this is to discover whether the software has any weird behavior that will be against the requirement of the software. If there is a weird behavior, it will be reported to the developers and the developers team will need to fix the problem until it met the requirement before it is ready to be deployed.

The last phase will be about deployment and maintenance. After the testing phase are successful, the software will be ready to be deploy and let the customer use. Once the software is deployed, there will be a maintenance team to take care any problem that the customer encountered.

**Waterfall Model**

Here is some example of the development models. First is the waterfall model, basically in this model every phase can only begin when the previous phase had been completed. This are all the phases in waterfall model, requirements, analysis, coding, testing and deployment. In every phase of this, what it does is the same as what I had mentioned above for the phases.

The advantage of using this model is that task is divided therefore it is much easier to deal with one task at a time. And since it is divided, in each phases the objective are well defined to produce the expected results. Waterfall model are usually used in small project where requirement is very clear.

The disadvantage of this is that the product will only be completed at the end and there will be no chance for user to test the requirement to check whether if that is what they wanted, therefore if there are changes needed to be made it will not be able to do so.

**Prototyping**

Prototyping is where a prototype is built first instead developing the system straight away. The prototype will then be refined until the targeted system. The goal of this model is to find out the requirement from the client. This model is used when the system has a lot of interaction with the end users.

There are few types of prototyping, the first one is the throwaway prototyping. What throwaway prototyping does is that, it has minimum knowledge of what are the required requirements and base on these requirements the prototype is built for the client. Then base on the client review, the specification will be written out and the current prototype will be throw away. This cycle will keep on repeating until the client are satisfied with the prototype. After the client are satisfied, the product will be built based on the specification.

The second type of prototyping is evolutionary prototyping. The evolutionary prototyping begins by building the prototype where part of the requirement is clearly understood. Once the prototype is done, let the client review it and add on any unknown requirement that we don’t know about it to let us continue to evolve the product. This cycle will keep on repeating until the client are satisfied and then the real product will be built.

The advantage of prototyping is that it helps us to clarify the user requirements and changes can be made based on client feedbacks. Any missing features can be identified easily as well.

The disadvantage of prototyping is there could be many iterations of the cycle which will make the project difficult. Since constant changes might cause the software structure to corrupt and this will make the changes more costly and difficult.

**Agile Methods**

In agile method, all the task is divided into small time frames to produce a certain feature for a release. Iterative approach is used to build the product and to be delivered in each iteration. In each build, it will be incremental in the features so that the product will have all the required features that the client want. Agile is good when new changes are needed to be implemented.

The advantage of agile methods is that this method uses interaction with the peoples. The client, developers and tester are communicating with each other constantly. The product itself also produced regularly in a much faster way. Constant attention to the technical excellence and good design. This method is also very good at adaption base on the condition. Changes can be made at later stage as well.

The disadvantage of this method is that it is difficult to evaluate the effort required at the start especially when the deliverables are very big. It lacks designing and documentation and the project can be off track easily if the client is clear of what the results they want.

**Reference**

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

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

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

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

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

<http://istqbexamcertification.com/what-are-the-software-development-life-cycle-sdlc-phases/>

<http://www.testingexcellence.com/software-development-life-cycle-sdlc-phases/>

\*Some resources are being used from the power point slides from lms.tp.edu.sg