Darren Teo

1506657D

**Question 1**

Software Development Life Cycle(SDLC) is defining task that are performed for every step in the software development process. SDLC consist of detailed plans on how to develop, maintain, and replace specific software. The life cycle defines a methodology for improving the quality of the software and overall development process.

There are several software development models followed by organizations.

SDLC consist of planning, Implementation, testing, documentation, deployment and maintenance and maintaining. Planning is the most important part of software development, requirement gathering or requirement analysis are usually done by the skilled engineers in the organization. Implementation happens when software engineers writes codes per the requirements. Testing is done to find bugs or defects in creating the software. Documentation is used for keeping every part of the projects for future references and to improve the project. Deployment happens when the software is ready to bring out to the market. Maintaining is done after the software is deployed. Software improvements and changes can take longer than the time needed to create the initial development of the software.

Waterfall model is one of the models that are used. This model would require each phase to be completed before moving on to the next phase. When each phase is completed successfully, it is checked if the project is on track and whether it is possible to continue with the projects. There a few stages for the waterfall model.

The first part for the model is to methodically analyse and written down in specification document that serves as basis for all future development. The result defines what the application should, but how it should be done. The second concept, the system is analysed to generate the models and business logics that will be used in the application. The next stage would cover largely on design requirements such as programming language, data layers, services, etc. A design specification will typically be created that outlines how the business logic covered in the previous stage which is technically implemented. The next part would include implementing all models, business logic and service integrations that were mentioned in the previous stages. For the next stage, the software would be tested by beta testers and other testers. They would discover and report issues. Finally, the application is deployed and there would be subsequent support and maintenance that may require to keep it functional and updated.

There are a few advantages to waterfall model. Waterfall model allows the project to maintain a wide scope and design structure due to planning and documentation stages. Waterfall model is suited for large teams that may see members come and go for the project, allowing the burden of the design to be placed on the core documentation which does not require much from team members. Waterfall model forces the project, and the project is disciplined in its design and structure. Waterfall model allows for early design changes.

There are a few disadvantages to waterfall. Waterfall model would ignore client feedbacks. Waterfall model would also include delayed testing period.

Agile model is another model that is used. Agile method breaks the product into small incremental builds. These builds are provided in iterations. Each iteration typically last about one to three weeks. Every iteration involves functional team working simultaneously on different areas such as planning, requirement analysis, Design, Coding, unit testing and acceptance testing. Agile model handles differently and the existing methods that fits the project requirements. In Agile model, the tasks are divided to smaller time frames to deliver specific features for release.

There are a few advantages of the agile model. The agile model promotes teamwork and cross training. Functionality can be developed fast and can be shown. The resources required are little, it is suitable for fixed or changing requirements.The agile model would deliver fast working solutions. It is a good model for environments that change steadily. There are very little rules in the agile model which would make documentation easily employed. Agile model is easy to manage and little or no planning is required. Agile model also gives flexibility to developers.

There are also a few disadvantages of agile model. The agile model is not suitable for handling complex dependencies. There is more risk of sustaining, maintaining and extensible for agile model. Agile model requires strict delivery management dictates the scope, functionality to be delivered and adjustments to meet the deadlines. If customer do not specify requirements properly, the team would be driven in the wrong direction. There is a very high dependency, since there is minimum documentation generated.

Prototyping is another model that is used. Prototyping is a working model of software with limited functionality. The prototype does not always have the same logic as the actual software and is an extra effort to be considered under effort Estimation. Prototyping allows to understand the user’s requirements which are user specific and is not considered by the developer during product design.

There are a few steps which is taken for prototyping. Firstly, understanding the very basics product requirements such as interface is important. At this stage, internal design is important and external aspects such as performance and security can be ignored at this stage. Next, the initial prototype is developed, where the basic requirements are showcased and user interface provided. Next, the product would be reviewed and feedback is collected and used for further improvements in the product.

Throwaway prototyping is one of the prototyping. It is also called the rapid or close ended prototyping. This type of prototyping use lesser efforts with minimum requirement analysis to build a prototype. Once the actual requirements are understood, the prototype will be discarded and the actual software will be developed with a better understanding of the user’s requirements.

Evolutionary prototyping is one of the prototyping. It is based on building actual functional prototypes with minimal functionality in the beginning. The prototype is based on actual prototypes that have minimal functionality at the start. The prototype would upgrade using the previous prototypes. By using evolutionary prototyping, requirements will be well-understood and included in the prototype.

There are also advantages of prototyping model. Prototyping model requires increased user involvement in the product even before it is implemented. Since a working version is displayed, the users get a better understanding of the system that is being developed. Prototyping model would reduce time and cost as the problems can be detected much earlier. Quicker user feedback helps to give better solutions. Missing functions can be identified easier. Confusing and difficult functions can be identified.

There are also disadvantages of the prototyping model. Prototyping model have a risk of insufficient requirement analysis due to too dependent on the prototype. Users may get confused in the prototypes and the real software. Prototyping might increase the complexity of the systems as the scope of the system may expand beyond original plans. Developers might reuse prototype instead of actual system. The effort invested in prototypes might be too much if it is not planned properly.

**References**

<https://www.techopedia.com/definition/22193/software-development-life-cycle-sdlc>

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

<https://airbrake.io/blog/sdlc/waterfall-model>

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