**Chan Kei On**

**P01**

**SWEN Assignment 1**

**Definition of SDLC**

Software Development Life Cycle or SDLC is a series of phases that provide a model for development and lifecycle management of an application or software. It is a framework that consists of a detailed step by step plan on developing, maintaining and replacing software. The life cycle defines a methodology for producing software that is of high quality and with the lowest cost in a short period of time.

**Phases in SDLC**

There are 5 phases in the SDLC:

**1) Requirement engineering**

Requirement engineering is the most important phase in the SDLC. In this phase the project managers, stack holders and users will meet to determine the business requirements. Questions such as; Who is going to use the system? What data should be collected by the system? How will they use the system? These questions will help the project managers gather what kind of features or requirements the stack holders and users want.

**2) Analysis and Design**

Once Requirement engineering is done, this phase is to clearly define and document the product requirements. This is done through a Software Requirement Specification (SRS) document which consists of all the product requirements to be designed and developed during the project. This helps in helps in specifying hardware and system requirements and helps in defining overall system architecture. Afterwards, based on the requirements in SRS, the development team would need to Produce a representation of an entity that will later be built. Includes architectural design, user interface design or database design.

**3) Implementation**

In this phase, the actual development and coding of the product has begun. The programming language is chosen with respect to the type of software being developed. Developers would use programming tools such as compilers, interpreters and debuggers to generate the code. Different high-level programming languages such as C, C++, Pascal, Java and PHP are used for coding.

**4) Testing**

In this phase, the product is tested where any defects are reported, tracked, fixed and retested, until the product is reliable and addresses the users’ needs gathered during the requirement phase. All types of functional testing like unit testing, integration testing, system testing, acceptance testing is done as well as non-functional testing are also done.

**5) Deployment**

Lastly, in this phase the product is delivered to the customer for their use prior to official release. The customers would do beta testing and if any changes are required or if any bugs are caught, the customers will report it to the development team. Once those changes are made or the bugs are fixed then the final release will happen.

**Software Development Models**

**1) Waterfall Model**

The waterfall model is the first SDLC process model to be introduced in software development. It is a linear-sequential life cycle model, this means that progress is steadily going downwards through phases of software implementation, like a waterfall. This means that any phase in the waterfall model can only start once the pervious phase has been completed as there are no overlapping in the phases. Therefore, it is very simple to understand and use.

The advantages of the waterfall model are that it is simple and easy to understand and use. It is easy for project managers to manage the project using waterfall model due to the rigidity of the model as each phase has specific deliverables and a review process and phases are completed one at a time. In waterfall model, the process and results are well documented.

The disadvantages of the waterfall model are that it is not suitable for projects where requirements are at a moderate to high risk of changing as the waterfall model has very little flexibility in going back to the previous stages that have been completed. The waterfall model is also costly and required more time, in addition to the detailed plan. Lastly, there is no working software produced until late during the life cycle, this means that it involves a high risk of uncertainty where the customer has different views of how the software would work once shipped but it is too late to change it.

The waterfall method should be used when the customer’s requirements are well known, clear and fixed, product definition is stable and ample resources with required expertise are available freely.

Based on the assignment, our group did not choose the waterfall method as we needed an SDLC model that can accommodate in changes of requirements as our project would require us to update the customers of the product periodically, which leads to a moderate to high risk of changing requirements as customers might want or change certain features. The waterfall model is also expensive which might pose a problem as the hotel we are building the product is not doing too well, meaning the budget for the development of the software is constrained.

**2) Big Bang Model**

The Big Bang model is an SDLC model where it does not follow any specific process and customer’s requirements are uncertain. The development just starts with focusing all possible resources in the software development and coding, with very little or no planning. Requirements are understood and implemented as they come without much analysis. Any changes required may or may not need to revamp the complete software. Big Bang model is used in small projects with one or two software engineers.

The advantages of the big bang model are being that it a very simple model making management of project easy, it requires little or no planning, very few resources are required and gives flexibility to developers.

The disadvantages of the big bang model are being very high risk and uncertainty, as there is little or no planning done as development starts immediately once requirements are understood. This means that the project can run into a potentially serious problem later down the line once deep into the project, which may cause the project to be cancelled or to be redeveloped, it also means the software might not represent what the customer wants. The project to get very expensive once requirements are misunderstood.

Based on the assignment, our group did not choose the Big bang model as the project we are assigned is complex and large enough that the big bang model is an insufficient SDLC model to use. As we need to periodically update the owners of the hotel, we are required to do show how the planning of the project is going, what features are going to be implemented and the budget for the development.

**3) Agile Model (Scrum)**

Scrum is an iterative and incremental agile framework. In Scrum, it starts when the customer requests a bunch of features they want in a product. These features are called user story as they are written from the perspective of the customer. All the user stories are added into a product backlog which is a wish list of all the features the customer wants. The development team then chooses the user stories they want to put into the product. The chosen user stories are added into a release backlog which the development team then prioritizes the user stories and estimates the story points and amount of time needed to complete each user story. Story points are a unit of measure for expressing an estimate of the overall effort that will be required to fully implement a product backlog item or any other piece of work. A story point includes the amount of work to be done, the complexity of the work and any risk in doing the work. All the estimates of each user story would be added to estimate the total amount of time needed to complete the entire release. Afterwards the user stories will be spilt into several sprints and these sprints are added into a sprint backlog. Sprints are short duration milestones ranging from 2 days to 4 weeks that allow the development team to complete a manageable portion of the project that is fully tested with all the features of the sprint and get it to shippable state to the customer. An important part of Scrum is daily scrum, these are daily 15 minute meetings where team members quickly list the work they have completed and any obstacles they have encountered. This ensures that team is always updated and any major issues are dealt with quickly. Once the sprint is complete, a sprint review is held to demonstrate what they have done to the customer by giving a demo of the product and its value. Afterwards, a retrospective meeting is held in which the team reflects on what when well and which areas need improvement. The cycle is repeated until a deadline arrives, enough items in the product backlog has been completed or the budget ran out.

There are three important roles in a development team using Scrum. Firstly, a product owner is someone who makes sure that the right features get added into the product backlog representing the users and customers of the product. Secondly, a Scrum Master is someone who ensures that the project is running smoothly and every member of the team has the tools they need to get the job done. The Scrum Master sets up meetings, monitor the work being done and helps with release planning. The last role is the development team which helps build the product and test it.

The advantages of using Scrum are that it emphasis on frequent updating of the progress in work through regular meetings. Thus, there is clear visibility of the project development. Scrum requires continuous feedback from the user which ensures that requirements are understood by both user and development team and there are no misunderstandings. Due to short sprints and constant feedback, any changes that need to be implemented would be easier to cope as it can be shifted to the next sprint cycle. Issues are identified in the daily meetings can be resolved quickly.

There are several risks involved in using Scrum, the most notable risk is scope creep as projects that uses Scrum does not have a definite end date, customers might be tempted to demand new functionality to be delivered. This would lead to the project taking a longer time to complete and the team might go over the budget that was given to them.

Another risk is high pressure or stress on team members as it requires the members to finish the work assigned to them in a strict and short deadline. This might lead to team members not being committed to the project, which could lead to the project to fail. If the task is not well defined in terms of cost and estimated time, the task will be spread over several sprints which will prolong the project.

Based on the assignment, our group chose the Scrum framework as it provides a high visibility in project development through Scrum meetings which tells the Scrum Master if the project is on schedule. As we need to periodically update the owners of the hotel, Scrum is a good framework to use as it requires continuous feedback from the user. This ensures that any requirements are understood and any changes that need to be done can be easily added into the next sprint cycle, making our project flexible. After every sprint, we are potentially able to deliver a shippable product for the owners to use which means that the owners of the hotel can start earning revenue.