CHAPTER 3: System Methodology

Every system must follow a specific system development methodology. The system methodology refers to as a standard process followed in an organization to conduct all the steps necessary to analyze, design, implement, and maintain information systems. The system must be developed according to the methodology so that the system and the documentation will not be mixed up and everything will get confusing. The use of a practical and suitable methodology are important because if there is any problem occur, enhancement to the system or changes that need to the system’s environment can be applied quickly and without many problem.

Considering all the steps in the system methodology, System Development Life Cycle (SDLC) is the most suitable methodology used in the system development. SDLC refers to the process of creating or varying systems, and the models and methods that people used to develop systems. The concept of it generally refers to computer or information systems. SDLC adheres the significant phases that are necessary for the development of a system. Those phases are planning, analysis, design, coding, testing and implementation as well maintenance.

Waterfall model is a classic model in SDLC, and the most common model used by developers because it utilizes a linear framework. In developing the application’s system methodology, such model is found useful and give a proper conduct to the system development. The Waterfall model runs through the process of analysis, requirements specification, design, implementation, testing and integration and lastly, operation and management. The flows are best described in the figure below.
Waterfall model is a chronological software development process, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. Basic principles of the waterfall model are separating the project into sequential phases which some overlap and splash back are acceptable between the phases. It emphasizes on planning, time schedules, target dates, budgets and implementation of an entire structure at one time. The Waterfall model also depends on the principle that a firm control is maintained over the project. In maintaining the control, it uses widespread written documentation and formal reviews as well approval signoff from the user. There is an occurrence of information technology management at the end of most of the phases before proceed to the next phases.
3.1 System Analysis

This phase deals with the requirements of the image encryption application. It records the problems that are faced by the end to end users. As the developer tried to resolve and analyzes the requirements, it provides basis acquirement of the resources. The resources are needed to achieve a resolution in the end. Due to this issue, specific system goals, punctual timeline and scope of areas of the project have been established. The system planning includes the range of the objectives for the system, justification of the project scopes, and presentation of possible solutions. The planning also offers potential problem faced, discovery of suitable resources like journals and summarizes them under literature reviews, and setting up the rest of the system development’s phases.

3.2 System and Object Design

The system and object design are to define the architecture of the application. In this phase too, the requirements of the system in the prerequisite analysis phase is mapped to structure the architecture of the system. The previous phase in the system is the “WHAT” phase, in the latter phase which is in the system design phase it is defined as “HOW” phase. This is where the development tools are defined. The type of programming editors that is being use, the environment of the system, the application architecture, platforms, data structures and many more are related to the design of this image encryption application.

The application later translated to logical and physical system specifications. The imperative works in this system design phase are designing the functional architecture of the system, sketching the interface layout of system, designing the flow of data by sketching the flow chart diagram and setting up the control of system.
It is in this phase too that the coding for the system is developed. Since the application built must provide ease of use in terms of portability, Java language is used to support cross-platform. The types of image files that are used in by the user to encrypt are taken into concern because every little aspect in developing the encryption is in a way affects the result of the system.

3.3 System Implementation

For the system implementation phase, the system developer function as to check the system using tools like debugger, emulators, compilers or interpreters. At this stage most of the components in the application are built. The components in the application are built precisely according the documents from the design phase and the requirements documents from the analysis phase. There should always be space for enhancement or updates.

Throughout the completion of the application there might be serious errors which can put a stop to the whole system’s flow. Therefore, in completing the application a fulfilling and satisfying the system of the system architecture requirement was vital.

3.4 System Testing

After finishing the implementation phase comes the testing phase. This is where the error found (if any) in the previous phase was being fixed. Testing phase also where the system is methodically verified to ensure that the system is error free and fulfilled the requirements outlined in the first phase.

There are three types of testing which is unit testing of individual code modules, system testing of the integrated system, and acceptance testing. Unit testing is where testing of the interactions of few functions but restricting the testing within one unit. Doing a test
code might be essential to support an individual test. Next, system testing is where all the
individual integration unit of the application is placed into a cohesive system and is tested in
the real-time environment, an actual computer application. Lastly, acceptance testing is where
the test is conducted by end users. The end users in the project are actually the application
users. If they find any flaws within the system, the end users will provide feedback to the
developers for corrections to be made in the future.

3.5 System Deployment

This phase is executed when the system has been tested and proven fit for use. If it is
proven, the developer can prepare the installation and user manual for user ease. In preparing
the application, this phase was the crucial one. This is because the application needs to go
through several testing where trial and error are a usual outcome. However, the outcomes of
every testing give a better insight on the deployment of the system. Only after the application
tested is successful that the phase is executed at the end.

3.6 Maintenance

In this phase it includes making alterations to the system or an individual component
to alter characteristics or having improvements to the system’s performance. The alteration
mainly found when the application is going through testing or being applicable to the user in
the system deployment phase. The alterations take place when request made by the users or
the flaws are found during live use or the system. The changes made to the application during
the maintenance phase are recorded in the documentation. Some of the changes that has been
recorded are in altering the Java developments, upgrading and improve the hardcoded
formula and adding the extra function to the application.
3.7 Advantages and Disadvantages of Waterfall Model

There are some reasons why Waterfall model is chosen to be used in the system methodology as it offers advantages that are suitable in developing the application. The staged development cycle enforces discipline which makes the application built in correct manner and step. This makes it easy and systematic as the process of developing any applications need a proper conduct and comprehensive process.

In using the model, every phase has a define start and end point in which progress can be conclusively indentified by both vendor and client. In this application, it involves the system developer and the end user. This is in a way provides a systematic step in creating the application. Not just that, it is also emphasizes on requirements and design which help to improve quality of the application beneficially.

As a student, time is important and because the development of the application needs a proper attention, Waterfall model gives a potential opportunity to minimize the time constraints. There is a lot of effort needed to bring a powerful outcome of the project. The model offers minimal wastage of time and effort as well to reduce the risk of schedule slippage. As what has been mentioned above, the model comprises a step by step phase which makes the system development in a sequence manner.

In the system and object design phase, a proper design of the application is needed. This stage has benefited the application in many ways because it is much easier to catch and correct possible flaws at the design stage than at the testing stage. In this stage that much of the flaws of the designed application found and immediate correction has been taken.

Each phase of the Waterfall model has specific deliverables and a review process. This make it is easy to manage the development of the project because of the rigidity of the model. It is important to note that the thesis application is a small project. The Waterfall
The model actually works well for smaller projects in which requirements for the application are very well understood. The model also gives advantages to the application development because testing is inherent to every phase.

Despite all the great advantages that the Waterfall Model offers, there are some disadvantages to the model. Not many systems are developed according to its sequential flow any more. There are disadvantages that indirectly affect the system developments. While every cycle is actually in discipline, adjusting scope during the life cycle can kill a project. Thus, in the development of the application, a proper adjustment of the scope was taken not to kill the project.

The Waterfall project advocates that no working software is produced until late during the life cycle. However, to overcome this procedure other works as such documentations and other developments of the application that are not related to the cycle are being done not to waste any valuable time as well to prepare any flaws when the software is produced later in the cycle.

The Waterfall model also is spotted to have a poor model for complex and object-oriented projects, long and ongoing projects and in project that requirements are at a moderate to high risk of changing. Another disadvantage of the model found when users don't really know what they want up-front; rather, what they want emerges out of repeated two-way interactions over the course of the project. Thus, specific up-front and outcome interactions have been planning earlier to overcome this problem to arise.

In every model used in any system methodology has its own pros and cons. Thus, as a researcher, there is a need to identify all of the advantages and disadvantages of one’s model. In using Waterfall model, a proper study has been done earlier to suit all the requirements needed for the system methodology as well to deny the disadvantages of the model.