

A Study on Software Development Life Cycle & its Model

Navita

Department of Computer Science,
Vaish College, Rohtak, Haryana, India

Abstract— Software development life cycle (SDLC) is a well defined and systematic approach. It practiced for the development of a reliable high quality software system. There are many SDLC models available. This paper describes five SDLC model , namely ;waterfall model ,iterative model, v shaped model ,spiral model, prototype model. Each development model has their advantages and disadvantages. The aim of this paper is to present a study on software development life cycle (SDLC) and SDLC models with their advantages and disadvantages.

Index Terms— *Advantages, Software Development, SDLC, V Model, Waterfall Model.*

I. INTRODUCTION

Today every work is done through technology. So, there is a need to develop more big and complex software system which meet the user's changing and growing requirements. So, SDLC provides planned and systematic arrangement of activities to be carried out to deliver high quality products within time and budget. SDLC is a methodology for designing, building and maintaining information and industrial systems. A software development process has different phases. These phases follow a top-to-bottom approach. Each phase takes input from the previous phase, add features and then produce outputs. The various phase of systematic development of software are shown in figure 1[1].

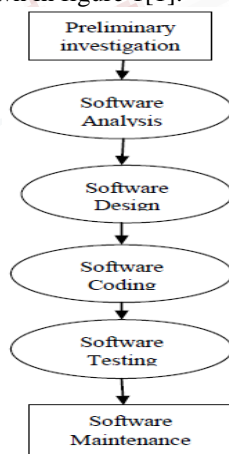


Fig. 1: Software Development Life Cycle.

Software development life cycle starts with preliminary investigation. This phase verifies the problem in present system and understands the need for the required system. The output of preliminary investigation decides whether the new system should be developed or not. After preliminary investigation, the output delivers to software analysis. In software analysis phase, studies the problem or requirements of software in detail. The main objective of this phase is to identify exactly "what type of features" the software to be developed must provide. Software analysis includes two distinct activities:

- Requirement analysis -: In requirement analysis, all related data and information of customer requirements are collect and analyze.
- Software requirement specification -: After analyzing requirements of the user, software requirement specification (SRS) is developed. In SRS, plans for project is develop which describe the activities to be performed during the project.

After software analysis, software design takes place, in this phase; the requirements are given a 'defined' form. After designing phase comes the coding. Coding is translating the design structure into a programming language [2].

Output of coding phase deliver to system testing, in this phase, ensures that the software is free from errors. After testing the whole software, software maintenance is required. The objective of software maintenance is to make the software operational as per user requirements

and to provide continuity of services. Using the SDLC phases the user can develop the software according to their requirements. This section has provided the phases of SDLC. In section 2, SDLC models are discussed. In section 3, the conclusion is presented.

II. SOFTWARE DEVELOPMENT PROCESS MODEL

A process model can be defined as a method, tool or steps for developing software. Some of the models are:-

➤ **Waterfall Model**

Waterfall model is the simplest and traditional software process model. Waterfall model is mostly used for small projects. This model works in a linear order. In this model, the output of one phase used as the input to the next phase. So it is also called linear sequential model [3].

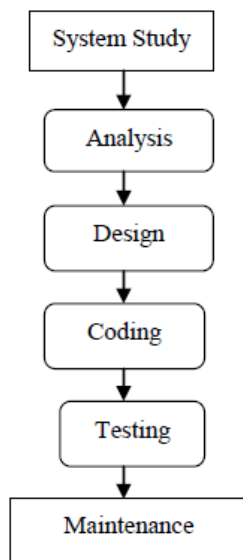


Fig.2: Waterfall Model

In this model, the first stage is system study. In this stage, studies the problem or requirements of software. After that in analysis phase, analyze the problem and requirements and in designing phase, design is made for requirements then in coding stage design structure is translate into programming language. Output of coding is delivering to testing phase. In this phase errors are removing from the software. After testing the whole software, Maintenance phase is required. Since these

steps are performed one after the other and fall like water falling from the sky that's why it is called waterfall model.

- ✓ Advantages
 - This model is easy to understand.
 - It is easy to explain.
 - Testing is inherent to every phase.
 - It is less costly.
- ✓ Disadvantages
 - In this model user training is less important.
 - Before system testing, problems are not discovered.
 - It is not useful for big projects.

➤ **Iterative Model**

This model includes the features of both waterfall model and prototype model. In this model the process start from the requirements and iteratively enhance the requirements until the final software implemented. In this iteration process, the delivery of the increments to the user continues until the software is completely developed [4].

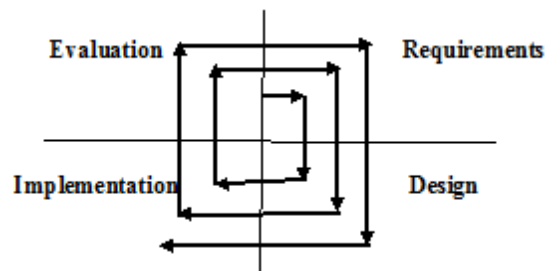


Fig.3: Iterative Model.

This model works on four phases -:

- Requirement phase -: In this phase, a list of requirements is prepared.
- Design phase -: In this phase, software is designed according to the requirements.
- Implementation phase -: in this phase, the software is implemented.
- Evaluation phase -: in this phase, the software is evaluated. The software is observed for further requirement.
- ✓ Advantages
 - In this model better testing is possible at each iteration.

- This model does not require high complexity rate.
- In this model feedback is generated quickly.

- ✓ Disadvantages
 - It requires planning of technical level.
 - It is not easily understandable.

➤ V Model

The v model is useful in every phase of the software development life cycle [5].

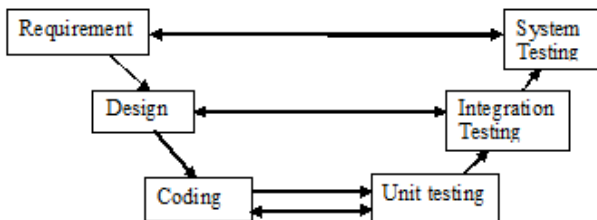


Fig.4: V Model.

The v model is divided into two branches, where the left branch analyses the requirement of the software to be developed. The right branch includes the testing activities. The left and right branches of this model work concurrently.

- ✓ Advantages
 - It is simple and easy to use.
 - It works well for where requirement are easily understood.
 - V model covers all functional areas
- ✓ Disadvantages
 - It is very inflexible.
 - It is costly and required more time.
 - This model doesn't provide a clear path for problem found during testing phases.

➤ Spiral Model

The spiral model is good for large, expensive and complicated projects. This model uses many of the same phases as the waterfall model. In this model, a spiral is used which has many cycles. Each cycle contains four

main activities represented by four sector- planning, risk analysis, development, and customer evaluation [6].

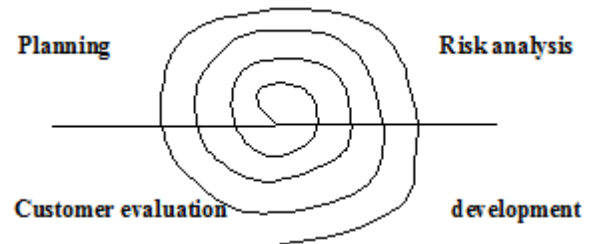


Fig.5: Spiral Model.

- Planning -: in this phase, objectives are determined and alternatives are identified.
- Risk analysis -: in this, alternative is evaluated, identify and resolve the risks.
- Development -: develop, verify next level product.
- Customer evaluation -: in this, customer evaluate the prototype.

Each cycle of this model is built the software progressively. In this all the risks are resolved and the software is ready for development.

- ✓ Advantages
 - It is more realistic model.
 - Each cycle of the spiral is completed by a customer evaluation.
 - It is good for large projects.
- ✓ Disadvantages
 - It takes more cost for development.
 - It is not good for smaller projects.
 - The success of project depends on the risk analysis phase.
 - It is time consuming.

III. CONCLUSION

In this paper, software development life cycle and SDLC models are defined. SDLC is helpful for user for getting a high quality product within time and budget. In this paper, five SDLC models are defined. Every model has its advantages and some disadvantage, like waterfall model

is good for small project and spiral model and prototype model is good for large and complicated projects. So that user can select the best suited model as per his requirement.

REFERENCES

[1] S. Maheshwari and D. C. Jain, "Comperative analysis of different type of models in software development life cycle," International Journal of Advanced Research in Computer Science and Software Engineering, Vol.2, Issue.5, pp. 285-290, 2012.

[2] M. Dahiya, "An Analysis of Life Cycle Models for CBSD," International Journal of Engineering and Technology, Vol.9, Issue.2, pp. 1044-1048, April-May 2017.

[3] Msritse 2012 Dr. W. <https://msritse2012.wordpress.com/2012/11/27/managing-the-development-of-large-software-systems/>.

[4] N. M. A. Munassar and A. Govardhan, "A Comparison Between Five Models Of Software Engineering," International Journal of Computer Science Issues, Vol.7, Issue.5, pp. 94-101, 2010.

[5] R. R. Raval and H. M. Rathod, "Comparative Study of Various Process Model in Software Development," International Journal of Computer Applications, Vol. 82, Issue. 18, pp. 16-19, November 2013.

[6] M. Batra and A. Bhatnagar, "A Comparative Study of Requirements Engineering Process Model," International Journal of Advanced Research in Computer Science, Vol.8, Issue.3, pp. 740-745, March-April 2017.