

A Study of Software Development Life Cycle Process Models

Shylesh S

Lecturer, Srinivas Institute of Management Studies, Mangalore-575002, India.

E-Mail : shylesh.s@live.com

ABSTRACT

The software development life cycle (SDLC) is used to design, develop and produce high quality, reliable, cost effective and within time software products in the software industry. This is also called software development process model. There are different SDLC process models are available. In this paper I have tried to describe different SDLC models according to their best use. There are many papers which have written in this regard. I will also use their knowledge or findings in this paper. The main purpose of this paper is to explain some of important SDLC models like Waterfall Model, Iterative Model, Spiral Model, V-Model, Big Bang Model, Agile Model, Rapid Application Development Model and Software Prototype. The main purpose of this paper is to explain advantages and disadvantages of these SDLC models. I will also describe which SDLC model is best fit for which type of software applications.

Keywords: Waterfall Model, Agile Model, RAD, Software Prototype

I. INTRODUCTION

All SDLC processes consist of a set of finite activities which are used to develop a software product. A SDLC process contains a complete plan for describing how to design, develop, maintain and increase the efficiency of a software product [1]. The SDLC process describes the methodologies which improves overall software quality and development process [2]. The following figure shows the different phases of a typical software development life cycle.

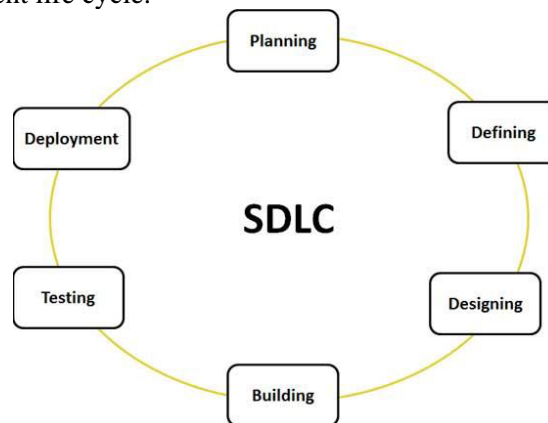


Fig. 1

We explain these phases of a typical SDLC process in a short and precise manner which is give below.

1) Planning

Planning and requirement analysis is the most vital and basic phase of every life cycle process. It is completed by the senior members after held meeting with customer or owner of the software system.

Quality assurance requirement, risk identification and feasibility report are the main outcomes of this phase.

2) Defining Requirement

When planning and requirement analysis is completed then the next phase is the detailed definition of the requirements, document these requirements and get verification from the customer. The output of this phase is the software requirement specification (SRS) document which contains all the requirements of product to be designed and developed [3].

3) Designing the Software Architecture

Software requirement specification is used as input to design the architecture of the software product which is being developed. Initially more than one architecture are designed and then reviews by important stakeholders according to criteria like risk assessment, robustness, design modularity, time and cost, the best design is selected [4].

4) Building or Developing the Product

In this phase, actual development of the product starts according to the designed architecture. If designing is done successfully then this phase is not much difficult. Developers use different tools such as compilers, interpreters and debuggers are used to generate code. Different programming languages like C, C++, Pascal, Java and PHP are used. Programming language depends upon type of software being developed.

5) Testing

In this phase developed product is tested whether it meets user's requirements which are documented in software requirement specification document. Software defects are reported, tracked, fixed and retested so that the product has gained high quality. Deployment and Maintenance After testing the product, it is deployed in its actual environment. Where customer verify his requirements which is called acceptance testing. After the feedback of customer, the further enhancement is done [5].

II. SOFTWARE DEVELOPMENT LIFE CYCLES MODELS

The most important and popular SDLC models are given below.

A. Waterfall Model

It is the first software development process model. Waterfall model is a sequential process model which does not overlap. It means that until the one phase is not completed then next phase cannot start. It is simple and easy to understand [7]. The graphical representation of water fall model is give below.

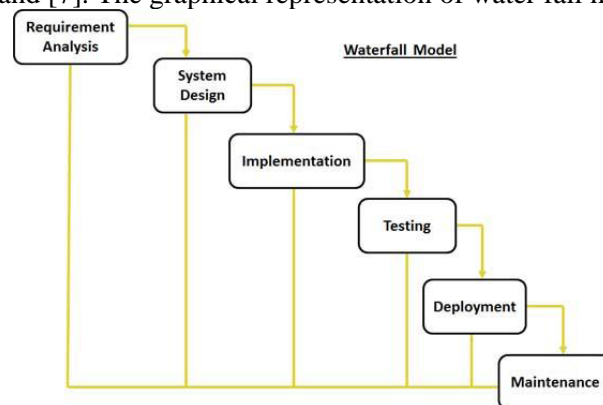


Fig. 2

The phases in this process model are:

1) Requirement Analysis

In this phase, all requirements of the software product are gathered in this phase and documented in software requirement specification document [7].

2) System Design

In this phase, overall structure of the software product is designed on the basis of requirement analysis phase.

3) Implementation

The development of software starts in this phase. It develops in small programs which are called units. These units are tested according to their functionality and integrated in the next phase.

4) Testing

In this phase, integrate the all units which are developed in implementation phase. After integration the whole product is tested to check whether it meets its goals. Software defects and bugs are reported, if they are available then fix and retested.

5) Deployment

When function and non-functional requirements are tested and validated then the software is deployed in the customer's environment.

6) Maintenance

If some problems are faced in the customer's environment then solve these problems in maintenance phase. Also some enhancement can do in this phase if user is not fully satisfied [8].

7) Waterfall Model Application

It is impossible that one SDLC model is fit for all types of software application. So it is very important to choose best software process model to specific software. I am tried to list some situations where waterfall model is best fitted which are

- Where software requirements are well understood, documented, cleared and fixed.
- Software definition is stable.
- Technology is understood and not dynamic.
- There is no ambiguity in software requirements.
- The required resources are available.
- The Project is small.

8) Advantages and Disadvantages of Waterfall Model

The advantage and disadvantage of waterfall model are given in the following table.

Table I

Advantages	Disadvantages
<ul style="list-style-type: none">• It is very easy to understand and use.[4]• Each phase has a specific deliverable and review process.[5]• Phases are processed and completed one at a time.• Works well for small projects where requirements are well understood.• Clearly defined phases.• Well understood deliverables.• Arranging tasks easily.• Process and results are documented.	<ul style="list-style-type: none">• Working software is available late during the life cycle.• It has a lot of risks.• A poor model for large projects.• It is not fit for projects where requirements are changed frequently.[4]• Difficult to measure progress during phases.• It cannot manage changing requirements.• Adjusting scope during the process model can end a project.

B. ITERATIVE MODEL

In iterative model, requirements are not completed and started iterative process with a small set of requirements. Each iteration evolves a small version of product and it is repeated until the final version is developed. Iterative process model starts implementation with a subset of requirement specifications. Each iteration is added new functionality in the process and continuous until it is completed [6]. The graphical representation of iterative model is given below.

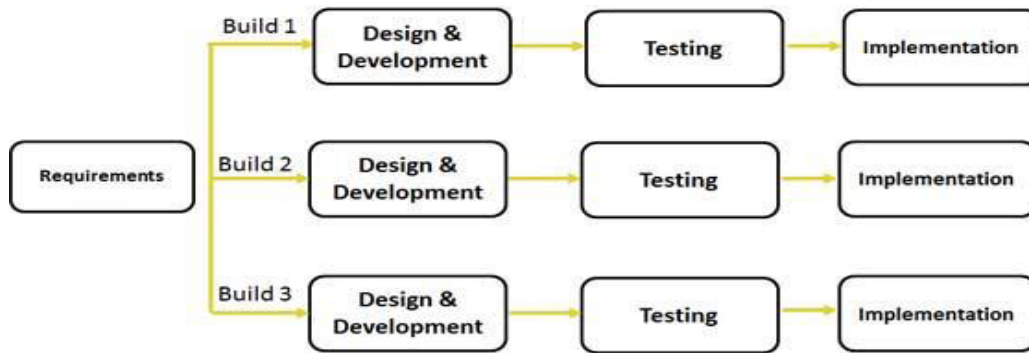


Fig. 2

1) Iterative Process Model Application

Like other SDLC model, Iterative model is also not fit for every application. However, this model is often used in the following scenario:

- When complete requirements of the system are cleared and well defined.
- Important and major requirements must be completed however some functionality may evolve with time.
- There is a time to market constrain.
- A new technology is being used while working on the project.
- Resources with needed skill set are not available and planned to be used on contract basis for specific iterations.

2) Advantages and Disadvantages of Iterative Model

Advantages and disadvantages of iterative SDLC model are shown in table given below.

Advantages	Disadvantages
<ul style="list-style-type: none"> • Some of important working functionality is available quickly. • Results are derived soon and quickly. • Parallel development can possible. • Project progress can be measured. • Requirement change process within budget. • In small iteration testing and debugging is easy. • Risk identification and milestone management is easy. • Most risky part is done first there easy manage high risks. • After every iteration functional product is delivered. • Issues, challenges and risk which get from one iteration are applied to the next iteration. • Better risk analysis. • It supports changing requirements environment. • Minimum initial operations. • It is best for large and critical products. • During iterative model software product is developed early which facilitates customer evaluation and feedback report. 	<ul style="list-style-type: none"> • Many resources are required. • Although requirement changing cost is low but not much appropriate for changing requirement. • More management is required. • System design issues are raised because not all requirements are available in starting. • Complete system definition is required for iterations. • Not best for small and tiny projects. • At the risk is unknowable. • For risk analysis and identifications, skilled persons are required. • Project progress depends upon risk analysis.

C. SPIRAL PROCESS MODEL

This SDLC model is the combination of iterative model and sequential model like waterfall model. The spiral model combines the idea of iterative and waterfall development in a very systematic and controlled way [6]. The diagram of spiral model is given below:

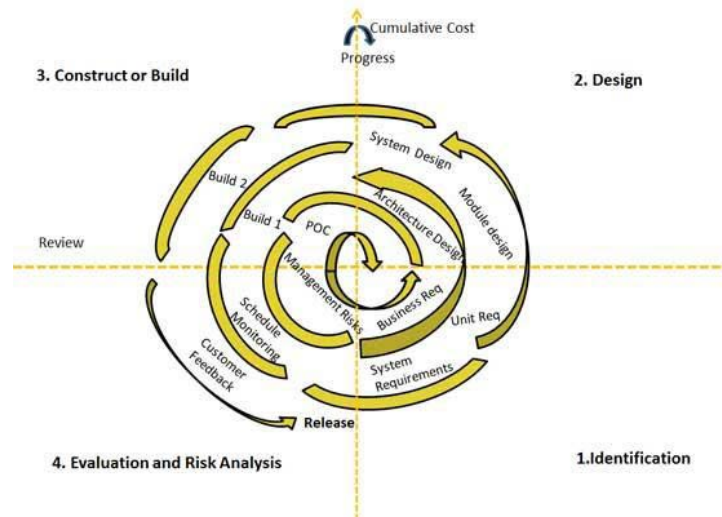


Fig. 3

Spiral model consists of four models which are:

1) Identification

The purpose of this phase is to collect business requirements. The system requirements, subsystem requirement and unit requirements are all done in this initial phase. For this purpose continuous communication with customer is done and at the end system is deployed in the customer's environment.

2) Design

In this phase, architectural design, logical design of modules, physical design and final design of subsequent spirals are included.

3) Construction

In this phase, actual development is started. In baseline phase only proof of concept is developed to get customer feedback. When requirements and design details are cleared then a working model is developed which is called build and send this build to the customer for further feedback.

4) Evaluation and Risks

In this phase, identifying, estimating and monitoring technical feasibility and management risk for example schedule slippage and cost overrun. The build developed in first iteration is sent to customer for evaluation and feedback. The next development starts based on the customer feedback in the next iterations. The iteration process continuous through whole life cycle.

5) Spiral Model Application

Spiral model is widely used in the industry. The situations in which spiral model are use are given below:

- Constraints on budget and risk evaluation are important.
- Medium to high projects.
- Where long project commitment due to changing requirement environment.
- When customer does not know his complete requirements.
- Requirements are complexe and need evaluation to get clarity.
- Where changes are expect in during life cycle.

6) Advantages and Disadvantages of Spiral Model

Table consists of advantages and disadvantages of spiral model is:

Table 3

Advantages	Disadvantages
<ul style="list-style-type: none"> • Continuous changing requirements can be managed. • Much use of prototypes is allowed. • Requirements collect more accurately. • Users see the system early. • Development process is divided into parts and risky parts are developed early which help better management. 	<ul style="list-style-type: none"> • Complex management. • Project completion duration is not known. • Not suitable for low risky and small projects. • Process is complex. • Spiral can go indefinitely. • Large number of phases requires heavy documentation.

III. CONCLUSION

This paper is about the different SDLC model and scenarios or situation in which these SDLC are best used. It can help project manager that which model is best for their projects. Also help developers, teachers, student and any other person which are interested in this topic. I have tried almost all popular SDLC models which are used in the software industry. I explained their advantages and disadvantages. Waterfall and Iterative Model is traditional and sequential model. Sequential means that the next phase will start only after the completion of first phase. Such models are suitable for projects with very clear product requirements and where the requirements will not change dynamically during the course of project completion. Iterative and Spiral models are more accommodative in terms of change and are suitable for projects where the requirements are not so well defined, or the market requirements change quite frequently. SDLC is a random approach to Software development and is suitable for small or academic projects. Rapid Application Development and Software Prototype are modern techniques to understand the requirements in a better way early in the project cycle. These techniques work on the concept of providing a working model to the customer and stockholders to give the look and feel and collect the feedback. This feedback is used in an organized manner to improve the product.

REFERENCES

- [1] Viller, S., & Sommerville, I. (2000). Ethnographically informed analysis for software engineers. *International Journal of Human-Computer Studies*, 53(1), 169-196.
- [2] Acuña, S. T., & Ferré, X. (2001, July). Software Process Modelling. In *ISAS-SCI (1)* (pp. 237-242).
- [3] Ballarini, D., Cadoli, M., Gaeta, M., Mancini, T., Mecella, M., Ritrovato, P., & Santucci, G. (2003, March). Modeling real requirements for cooperative software development: A case study. In *2nd Workshop on Cooperative Supports for Distributed Software Engineering Processes, Benevento, Italy*.
- [4] Munassar, N. M. A., & Govardhan, A. (2010). A comparison between five models of software engineering. *IJCSI*, 5, 95-101.
- [5] Krishna, S. T., Sreekanth, D. S., Perumal, K., & Reddy, K. R. K. (2012). Explore 10 Different Types of Software Development Process Models. *International Journal of Computer Science and Information Technologies*, 3(4), 4580-4584.

- [6] Tarkhala, N., & Kanani, C. (2014). IMPROVING THE SOFTWARE DEVELOPMENT EXPERIENCE BY REDUCING EFFORT, COST AND RESOURCE ALLOCATION BY COMBINING VARIOUS SOFTWARE DEVELOPMENT APPROACHES. *Development*, 1(11).
- [7] Shaw, M. (2003, May). Writing good software engineering research papers. In *Software Engineering, 2003. Proceedings. 25th International Conference on* (pp. 726-736). IEEE.
- [8] Gil-Garcia, J. R., & Pardo, T. A. (2006). Multimethod Approaches to Understanding the Complexity of e-Government. *Int. J. Comput. Syst. Signal*, 7(2), 3-17.
- [9] SVITS, I. M. (2012). A Comparative Analysis of Different types of Models in Software Development Life Cycle. *International Journal*, 2(5).