Chapter 27
Software Development Life Cycle Performance Analysis

Shilpa Sen, Mayank Patel, and Ajay Kumar Sharma

Abstract In 1960s SDLC (Software development life cycle) was initiated to develop high quality systems in large scale. Software development life cycle is one of the most liked techniques nowadays to develop a powerful and high quality software product. Here various types of methods present in software development life cycle like waterfall, iterative, prototype, v-shaped, spiral and etc. each method has drawback and benefit. According to any one of these models, the software will be developed. This method helps a lot to developer's team to develop a product in easy and faster way. This technique is referred to the decrement of problems and increment of value of the software project. The SDLC methodology is very hard without a sequence order process to develop a good quality product (Manzoor Ahmad Rather: "A Comparative Study of Software Development Life Cycle Models", IJACSA (2015).).

Keywords SDLC model · V-shaped · Prototyping · Difference · Disadvantage

1 Introduction

In the world of technology it is important to build a good quality product which meets user requirement. SDLC is the system development life cycle which helps us to manage and meet the product approach in the symmetric disciplinary way. It is usual procedure followed by software companies for understand requirement, design, implement, checking and evolution of product. Software engineering is referred to the quality priority, procedure, technique and devices that are used in process of developing a product. This process includes various kinds of task and activities and also this process gives the method to implementing a technique that covers requirement, design, implement, checking and evolution of product. SDLC is separated into different parts that allow any software company to handle the project in easy way. It is very famous and important technique develops software [1].

S. Sen (✉) · M. Patel · A. K. Sharma
Geetanjali Institute of Technical Studies, Udaipur, India

R. Mathur et al. (eds.), *Emerging Trends in Data Driven Computing and Communications*,
Studies in Autonomic, Data-driven and Industrial Computing,
https://doi.org/10.1007/978-981-16-3915-9_27

1.1 Phases of SDLC

There are various phases in SDLC given below (Fig. 1).

Requirement Analysis.

In this phase project developer's team make sure that they will gather all the requirements to client about project like user interface, functional requirement, framework selection, system technical architecture, and project planning [2, 3].

Design.

This phase is about finalizing design, user interface, requirement detail specification, and system technical architecture [4].

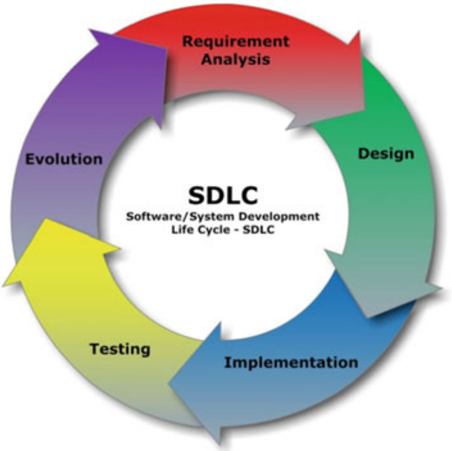
Implementation.

In this phase developers implement the code according to finalized design and before sending for the main testing of project they test the code that is called unit testing [5].

Testing.

This phase is the most important part of the software development life cycle because they check code quality and test according to requirements. After this phase project will be sent for deployment [6, 7].

Fig. 1 Phases of SDLC



2 Types of Models

2.1 Waterfall Model

In 1970 Royce proposed this model named as "waterfall model". The waterfall model is a popular form of the systems development life cycle model for software engineering [8]. The waterfall model describes a classic approach of development method as sequential and linear. Waterfall model has different approaches for each interpretation of development. It works like the output of each step in this model is the input of next step [9]. Once a development of interpretation is finished, development proceeds to the next interpretation and no overlapping happens. The interpretation includes Requirements definition, Design, Implement, Checking, and Maintenance [2, 10] (Fig. 2).

Requirement Analysis. All possible requirements are captured in requirement gathering documents.

Design. Based on requirement analysis software architecture design was finalized.

Implementation. Coding of the project in the little unit with its functional testing.

Checking. To make sure that the project is meeting customer requirements.

Maintenance. Fixing issues and release new version with the issue patches as required.

Advantages. It is understandable, simple and easy to use and easy to manage as each interpretation has its outputs and review process. It is well for smaller products where requirements are very well cleared.

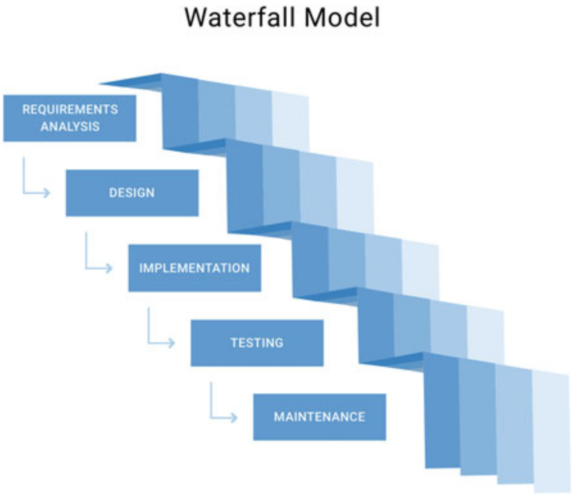


Fig. 2 Waterfall model

Disadvantages. When the project is checking interpretation, it is not possible to turn behind and update something which is not well. It is highly risky and uncertain method.

2.2 *The Iterative Model*

When some problems were seen in the first version of the model, then a new version of the model was created which was named as iterative model [2, 11]. It is the combination of both prototype model and waterfall model, and it is upgraded version of waterfall model. Iterative model comes when waterfall model is come up with a lot of drawbacks [12]. It splits the task into different fragments so that the product developing group can do their task faster and in an easy way. This model will work in the loop of Design, Implementation, Testing, and Review until software is deployed. This model is more flexible and it includes more features then waterfall model (Fig. 3).

When To Use Iterative Model

It uses when the product is big; the product's requirement is completely defined and the large requirement is clearly defined.

Advantages

It is faster, simpler, and easier to manage the classical model.

Disadvantages

This method is not well for smaller projects. It is not understandable and we cannot change requirements on multiple times.

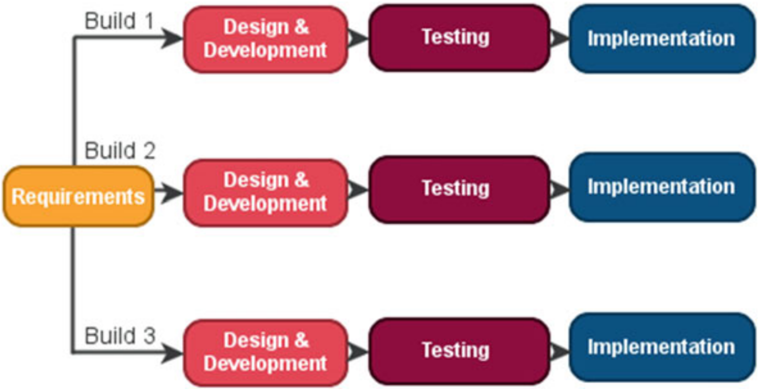


Fig. 3 Phases of SDLC

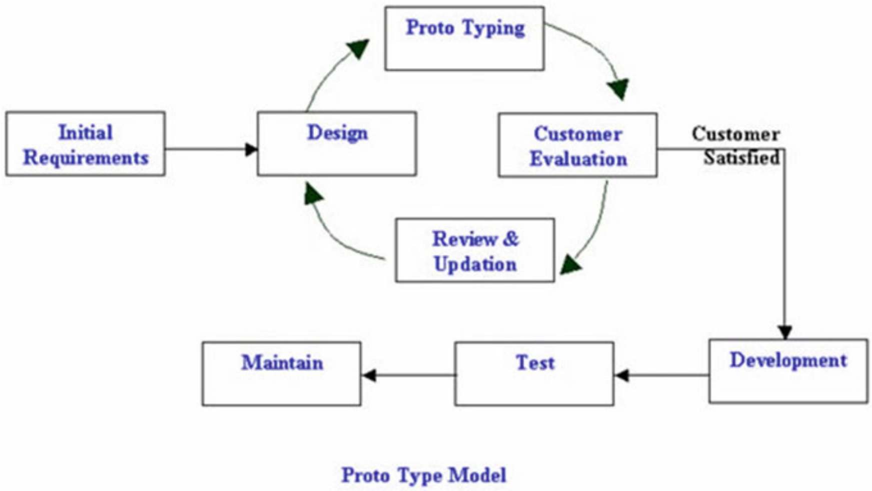


Fig. 4 Prototype model

2.3 The Prototype Model

Prototype means a kind of dummy also you can say a toy model of data which created according to client's requirement to develop a product [13]. This model is created only when the customer's ideas are not clear. This model is also called as "Throwaway model" because if the customer is not clear with their requirements then the prototype model will be created again. Then it will process to iterative model for design and developing a product (Fig. 4).

Advantages

This method is faster in design, its outcome is good, and it is simple to find bugs, it is understandable to developers.

Disadvantages

It is costly than other methods; it is not flexible and not good for big projects.

2.4 The Spiral Model

In 1986s Barry Boehm referred to this method. It is the most important model among all models in SDLC [14]. This model discards the mistakes of all models. Spiral model can analyze higher risk and also it can find an alternative solution to make better product without any errors. This model has four steps named as planning, Analyses of Risk, Design, Implement, Checking, and Maintenance. And its diagram

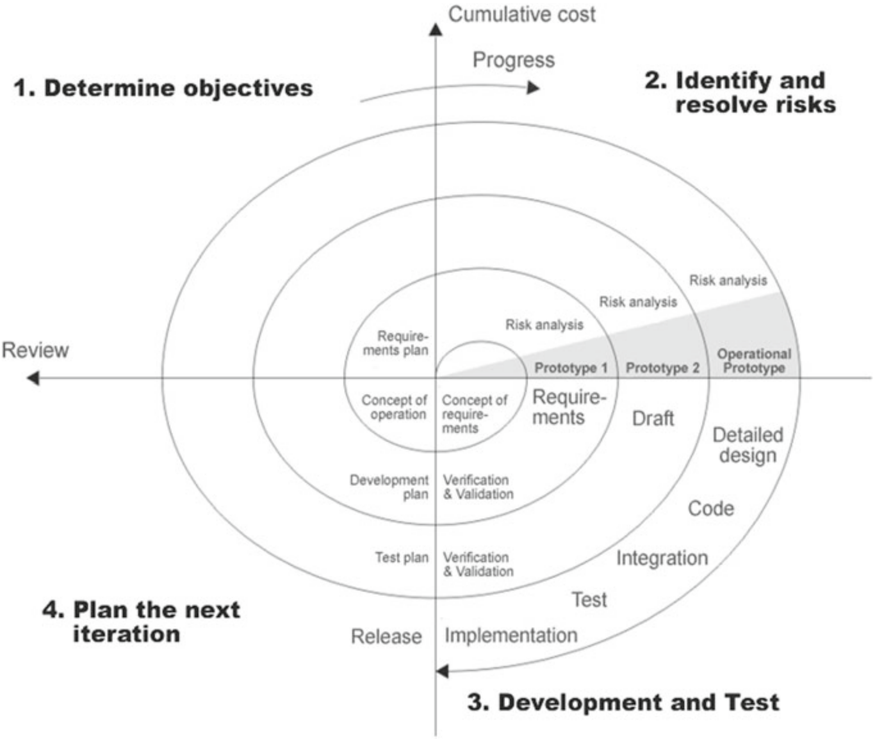


Fig. 5 Spiral model

(Fig. 5) looks like a spiral. This model is used when the product is large and has more time duration to complete the product.

Advantages

It can easily handle risk and it is better for big products; it is flexible according to requirement; it takes less time to deploy the product.

Disadvantages

It is more complicated than all models of SDLC; it is not good for small products and it is not easy in managing time during development.

2.5 V–Model

V-model is much-disciplined systems development life cycle model for software engineering which also can be considered an upgrade version of the waterfall model [15]. V-model describes relationships between the testing stage which is parallel to

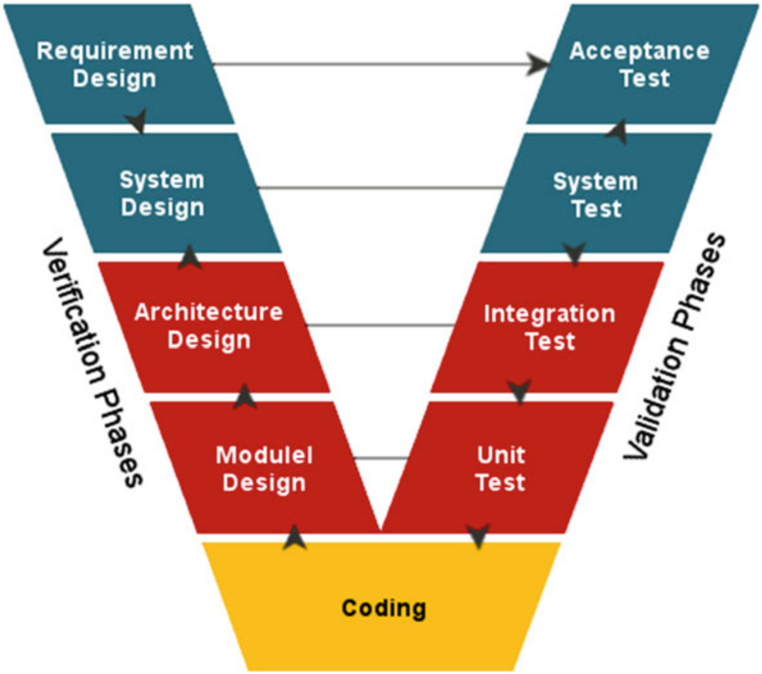


Fig. 6 V–Model

each development stage, i.e., In V-model testing is done on each interpretation that is parallel with development in a steady way. It is known as the Verification Model or Validation Model (Fig. 6).

Advantages

It is easy, understandable, simple and better for big products and it is easy errors.

Disadvantages

It is not flexible, more complex, not frequent, and highly risk; it is very high in cost and not good for small products.

3 Difference Between Models

See Table 1.

Table 1 Difference between all models

Model	The waterfall model	The iterative model	The prototype model	V-Shape model	Spiral model
Payment	Not high	Not high	High	Very high	High
Time	Long	Less	Less	Less	Flexible
Elasticity	Low	Low	High	Low	Flexible
Safety	Low	Limited	Low	Limited	High
Evaluation of risks	Only initially	Low	High	Low	Low

4 Conclusion

Since 1960 still today SDLC has made the work of project handling team easy to build software. All models of software development life cycle have their own advantages and disadvantages [1]. Software development life cycle brought new forms at every level and updates at each level with some advantages and disadvantages. By these different types of models a developer's team can do their work easier and faster. Main conclusion of software development life cycle models is to simplify the process of developing a software project easy, faster, and with less risk.

References

1. Rather MA (2015) A comparative study of software development life cycle models. IJACSA
2. Navita (2017) A study on software development life cycle & its model. IJERCSE
3. Boehm B (2000) In: Hansen WJ (ed) Spiral development: experience, principles, and refinements
4. Pressman R, Software engineering: a practitioner's approach, 6th edn. McGraw-Hill Publication
5. Aggarwal KK, Yogesh Singh Software Engineering, 3rd edn
6. Karlm (2006) Software lifecycle models, KTH
7. Rastogi V (2015) Software development life cycle models-comparison, consequences. Int J Comput Sci Inf Technol 6(1). ISNN: 0975–9646
8. Maheshwari S, Jain DC (2012) Comperative analysis of different type of models in software development life cycle. Int J Adv Res Comput Sci Soft Eng 2(5):285–290
9. Munassar NMA, Govardhan A (2010) A comparison between five models of software engineering. Int J Comput Sci 7(5):94–101
10. Raval RR, Rathod HM (2013) Comparative study of various process model in software development. Int J Comput Appl 82(18):16–19
11. Ben-Zahia MA, Jaluta I (2014) Criteria for selecting software development models. In: 2014 global summit on computer & information technology (GSCIT). IEEE
12. Velmourougan S et al (2014) Software development life cycle model to build software applications with usability. IEEE
13. Velmourougan S et al (2014) Software development life cycle model to improve maintainability of software applications. IEEE

14. Thitisathienkul P, Prompoon N (2014) Quality assessment method for software development process document based on software document characteristics metric. In: 2014 Ninth international conference on digital information management (ICDIM). IEEE
15. Davis A.M, Bersoff E. Hm Comer E.R, A strategy for comparing alternative oftware development life cycle models. In: IEEE transactions on Software Engineering, vol 14, Issue 10. Publication Year: 1988