

Review Paper

A Comprehensive review on Software Development Life Cycle

Gargi Kalia¹

Department of Computer Science Engineering, Universal Group of Institutions, India

Correspondence should be addressed to Gargi Kalia; gargikalia6@gmail.com

Received 02-12-2020; Accept 21-12-2020; Published 24-12-2020

Handling Editor: Sparsh Sharma

Copyright © 2020 Gargi Kalia. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Software Engineering is a branch of computer science which deals with software production. Software engineers develop different kinds of software's and provide proof to validate these results. This paper basically gives a review of software development life cycle. It gives a detailed insight of all the models used in software engineering such as Classical waterfall model, Spiral model, Prototype model, Big-Bang model, Incremental model, Agile model and V-Shaped model. Software Development Life Cycle (SDLC) is a procedure used for designing, developing and testing software's. SDLC helps in providing best quality software's which meets customer's expectations and it ensures the timely delivery of the project.

Keywords: *Software Development Life Cycle, Software Engineering, Incremental Model, Waterfall Model, Prototype Model, Spiral Model, Big-Bang Model, Agile Model, V-Shaped Model*

1. Introduction

Software is a collection of routines or subroutines which tell the computer how to work. Software is fuelling the modern era. Software consists of set of procedures, routines or subroutines which is linked with the operations used in computer system. Software Engineering deals with the

applications which are implemented with the help of science and technology [1]. It also explains the software approach development, operation and maintenance of software [2]. It interacts with the engineering branch which is concerned with all particulars of software development [3]. It is an engineering

discipline which establish the engineering principles so that one can obtain a reliable software which works efficiently [4]. It is a branch of computer science which helps to design, implement and maintain complex computer programs [5]. Software Engineering helps in activities used in computer programming and system analysis [6]. Software Engineering also deals in field of computer programming [7]. It can be considered as the term which is used to defend the codification of recommended practices used in engineering disciplines [8].

2. Software Development Life Cycle

Software industries utilize SDLC to design, develop and test excellent quality software. The main aim of SDLC is to build software which are of excellent quality. It builds a software which meets the customer's expectations and completes on time and within given cost estimates.

2.1 Software Development Life Cycle Models

Another name for SDLC is Application Development Life Cycle. SDLC has different phases such as Requirement gathering and analysis phase, Feasibility study, Designing, Coding phase, Testing phase, Deployment phase and then Maintenance phase.

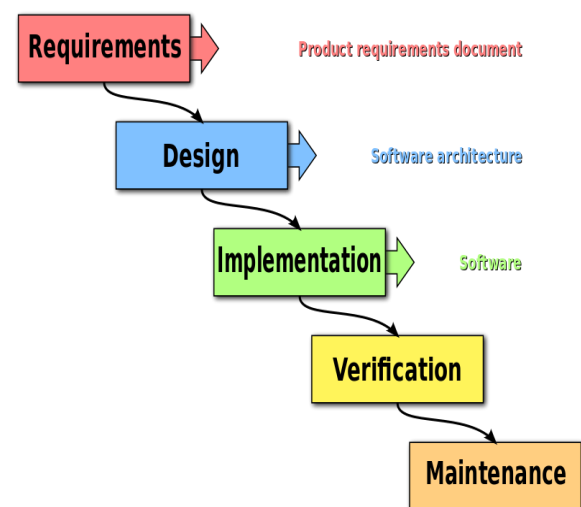
Software Development Life Cycle has various models such as:

- Classical Waterfall Model
- Incremental Model
- Spiral Model
- Big-Bang Model
- Agile Model
- Prototype Model
- V-Shaped Model
- Spiral Model

- Big-Bang Model
- Agile Model

2.2 Classical Waterfall Model

Classical Waterfall Model is easily understandable and manageable. In this model, one phase reaches completion, after that second phase gets started. One cannot come back to the previous phase. Each phase depends on the information received from previous stage and has its own project plans. Waterfall model is oldest model. It is a straight forward model. In this model, the flow direction is from top to bottom. Waterfall Model is firstly introduced by Winston W. Royce in his article in year 1970 [9]. He did not use the term "waterfall" in his paper. He elaborated that model as a pattern which is not working [10]. The term "Waterfall" is introduced in an article in 1976 [11].



2.3 Incremental Model

The incremental model is also known as Iterative model. This model implements Waterfall model incrementally [12]. It combines the waterfall model with the iterative theories. In this model, after each phase a new version of software is produced. This process repeat itself, until

the complete product is delivered. The sequence of releases is known as “increments”. With each increment, a bit more is added each time. This process repeats itself until the software reaches its completion. With each increment, new functionality is added to the product. It depends on the customer’s feedback. Planning is done for the next increments and moderations are performed accordingly. A new version of software is developed with every iteration. The iterative model applies the waterfall model repeatedly until the final product is delivered [13]. The iterative or incremental philosophy is also applying in agile modelling. This model has one disadvantage that resources gets exhausted by repeating the process over and over again.

2.4 Prototype Model

In Prototype model, a prototype or dummy of a product is built and tested. This process continues until expected results are achieved. One can have in prospect that individual sample or prototype costs will be high as compared to the final product due to the inefficiencies in processes. Prototypes are helpful for reducing costs by optimizing and refining the prototype accordingly [14]. A running prototype of a software is produced before deployment of actual software. Firstly, requirements are gathered in prototype model and then analysis is performed. After that quick decision is being made. Prototype is constructed and then assessment of that prototype is performed whether it has been built correctly or not. Later, user evaluates whether the prototype is built as per the desired results or not. If not, then refinements of that prototype are performed.

2.5 Spiral Model

Spiral model is a model which is used to mitigate risk factor. Spiral model guides to use components of waterfall, incremental or prototype model. Spiral model is the most flexible model among other models. In this model, the project deals with four phases again and again until final completion is achieved. It considers multiple rounds for refinement. Spiral model works in four steps: (i) Determine objectives (ii) Identify and Resolve risk (iii) Development and Test (iv) Plan the next iteration. This model was firstly introduced by Barry Boehm in his article in 1985 [15].

2.6 Big-Bang Model

The Big Bang model do not follow any particular process. This model requires very less/no planning. This model is used for small projects and requires very small development team of around one or two software engineers. This model is not used for complex projects. Risk factor is also high in this model. If the requirements are misinterpreted in the starting then project have to be started all over again. Big Bang model is a viewpoint to Software development and is suitable for small or academic projects [16]. In Big-Bang approach, the entire software is delivered in one shot at the end [17].

2.7 Agile Model

Agile modelling is a method to create models and documentation of software systems based on best practices. This model breaks the product in to cycles. Agile model delivers working product and considers a pragmatic development approach. This modelling produce system with proceeding releases with iterative changes from the previous release. With each iteration, product is evaluated. In this model customers, developers and testers work together. Therefore, it gives stress on

interaction. The project can go in wrong direction if the customer is unclear about the way customer wants to go.

3. Conclusion

This paper gives an overview about software development lifecycle. Software engineers helps in developing different kinds of software and provide proof to validate these results. This paper basically covers software development life cycle models. It gives a detailed insight of all the models such as Classical waterfall model, Incremental model, Prototype model, V-Shaped model, Spiral model, Big-Bang model and Agile model. SDLC is a procedure used for designing, developing and testing software. SDLC helps in providing best quality software which meets customer's expectations and ensures the timely delivery of the project.

References

- [1] Liu, Y., Lin, J., Cleland-Huang, J., Vierhauser, M., Guo, J., & Lohar, S. (2020, September). SENET: A Semantic Web for Supporting Automation of Software Engineering Tasks, *IEEE Seventh International Workshop on Artificial Intelligence for Requirements Engineering (AIRE)*, pp. 23-32.
- [2] Bhatia, M. P. S., Kumar, A., Beniwal, R., & Malik, T, Ontology driven software development for automatic detection and updation of software requirement specifications. *Journal of Discrete Mathematical Sciences and Cryptography*, 2020, vol. 23, no.1, pp. 197-208.
- [3] Sommerville, Ian. "1.1. 2 What is software engineering?." *Software Engineering, 8th ed., Harlow, England: Pearson Education*, 2007.
- [4] Mahoney, Michael S. "The roots of software engineering." *CWI Quarterly* 3.4 (1990), pp. 325-334.
- [5] Boehm, Barry W. "Software engineering economics." *IEEE transactions on Software Engineering* vol. 1, pp. 4-21, 1984
- [6] Salah, Akram I. "Engineering an Academic Program in Software Engineering." In *35th Annual Midwest Instruction and Computing Symposium*. 2002.
- [7] Mills, H.D., Newman, J.R. and Engle, C.B., An undergraduate curriculum in software engineering. In *SEI Conference on Software Engineering Education*, (pp. 24-37), Springer, 1990.
- [8] Barbara Kitchenham, O. Pearl Brereton, David Budgen, Mark Turner, John Bailey and Stephen Linkman, "Systematic literature reviews in software engineering" – *A systematic literature review, Information and Software Technology*, Vol. 51, Issue 1, Pp. 7-15, 2009.
- [9] Royce, W. W. , Managing the development of large software systems, *proceedings of IEEE*, WESCON, pp. 328-388, 1970.
- [10] Rerych, Markus. "Wasserfallmodell> Entstehungs context." *Institut für Gestaltungs-und Wirkungsforschung, TU-Wien*. Accessed on line November 28 (2007).
- [11] Bell, Thomas E., and Thomas A. Thayer. "Software requirements: Are they really a problem?." *Proceedings of the 2nd international conference on Software engineering*, pp. 61-68, 1976.
- [12] Pressman, Roger S. *Software engineering: a practitioner's approach*. Palgrave macmillan, 2005.

[13] Pressman, Roger S. *Software engineering: a practitioner's approach*. Palgrave macmillan, 2005.

[14] Gschwind, Michael, Valentina Salapura, and Dietmar Maurer. "FPGA prototyping of a RISC processor core for embedded applications.", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* Vol. 9, no. 2, pp. 241-250, 2001.

[15] Boehm, Barry. "A spiral model of software development and enhancement."

ACM SIGSOFT Software engineering notes, vol. 11, no. 4, pp.14-24, 1996.

[16] Ali, Kazim. "A Study of Software Development Life Cycle Process Models." *International Journal of Advanced Research in Computer Science*, Vol. 8, no. 1, 2017.

[17] Kumar, Naresh, A. S. Zadgaonkar, and Abhinav Shukla. "Evolving a new software development life cycle model SDLC-2013 with client satisfaction", *International Journal of Soft Computing and Engineering (IJSCE)*, vol. 3, no. 1, pp. 2231-2307, 2013.