Analysis of Software Development Life Cycle Models

Amninder Singh and Puneet Jai Kaur

1 Introduction

Nowadays the computer technology has rapidly changed, everyone wants the best software system that is fast, cheap, bug free, maintainable, and time saving that can fulfill all demands. Such type of product needs a working scenario that covers all the aspect. Best way to create an application is to choose software development life cycle also known as application development life cycle. It creates software in systematic manner and assures the product meets the needs of customer. The products are economical and highly excellent. It provides better resource management and plan for workflow. There are five models based on SDLC: Water fall model, Incremental model, RAD model, Spiral model, V model, Agile model. There are two types of software development methodologies, heavy weight and light weight. There is no model best or worst among both type of methodologies in different circumstances. The selection of best methodology depends on its nature, skills of team members and management criteria [1]. In SDLC software usability is an important attribute to assure quality. Scarcity in usability cause loss in way of cost, trust, and reputation. For this purpose, a new technique called U-SDLC is introduced. It concentrates on key attribute of every phase that gives best quality product with less number of errors [2]. Software maintenance is always a big issue for SDLC models to overcome from problem there is a new model created that concern with maintainability tasks called M-SDLC. It gives best maintenance practices in each phase of SDLC. That helps to create products that need low maintenance [3]. Management of any project providing better supremacy and

A. Singh (✉) · P. J. Kaur
UIET, Panjab University, Chandigarh, India
e-mail: amninder.saini001@gmail.com

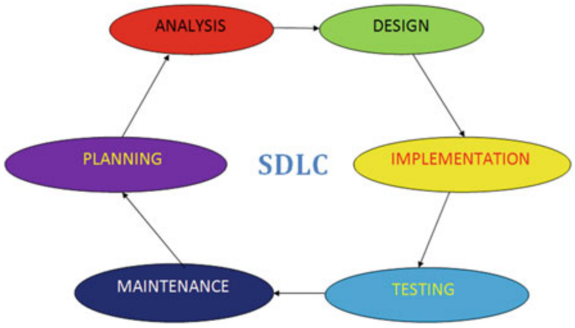
P. J. Kaur
e-mail: puneet@pu.ac.m

V. Nath and J. K. Mandal (eds.), *Proceeding of the Second International Conference on Microelectronics, Computing & Communication Systems (MCCS 2017)*, Lecture Notes in Electrical Engineering 476, https://doi.org/10.1007/978-981-10-8234-4_55

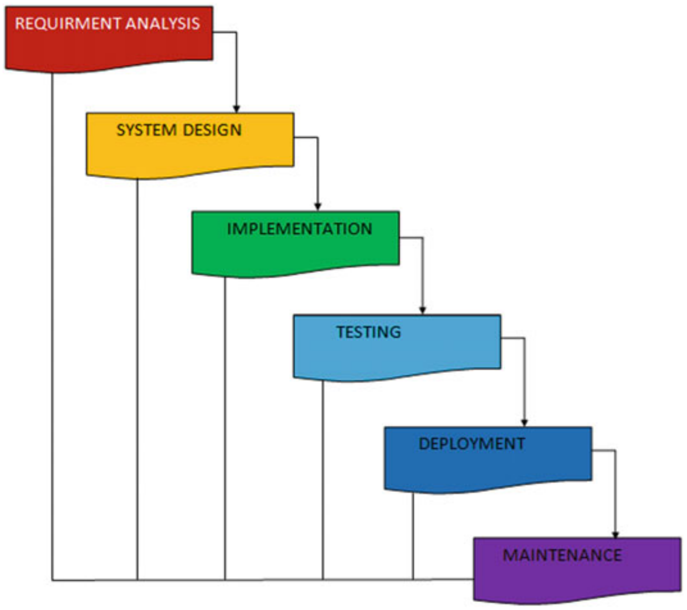
minimize the workload by diving complex task into small ones. But none of the SDLC models provide management for this purpose a new hypothetical SDLC model introduce that facilitate change, incident and release management. It creates three-dimensional model containing user, owner, and developer of the project. It helps to find more bugs and to cure them. It also gives better accuracy [4]. In this paper Sect. 2 contains description about various SDLC models. Section 3 is about analysis or comparison between models. The last section concludes all the paper's contents and at last references.

2 Software Development Life Cycle Models

SDLC: Software development life cycle model is a technique to create a software product that assures quality and easily fulfill the needs of the customer. It works in a systematic manner that is economical and efficient for the project management technique. There are six phases used in it that are Planning, Analysis, Design, Implement, Test, and Maintain.

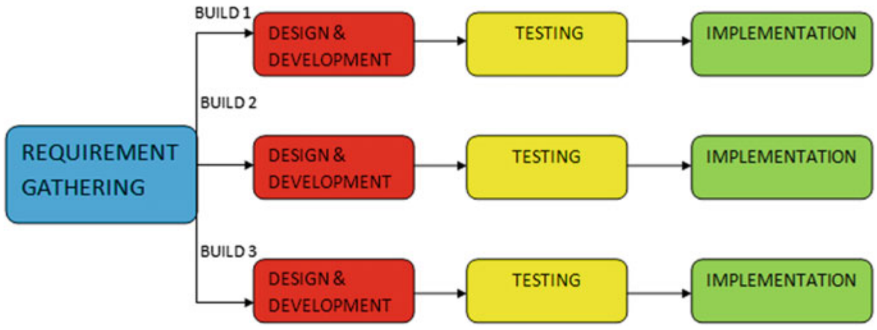


Waterfall Model: It is a linearly processed model, in which workflow stably goes downward. The model is processing in one direction only and very simple to use. Once a phase is processed it does not reuse again for further processing and each phase must be executed fully before next phase originates. Every other model considers it as a base model. After completion of every phase or stage, a review takes place to check if the project is on right way. If it does not show results as desired, taking a decision whether it to continue or eliminate the project is important. It is used for small project where all requirements should be clear. There are several phases introduce in waterfall model which are requirement analysis, system design, implementation, testing, deployment, and maintenance [5].

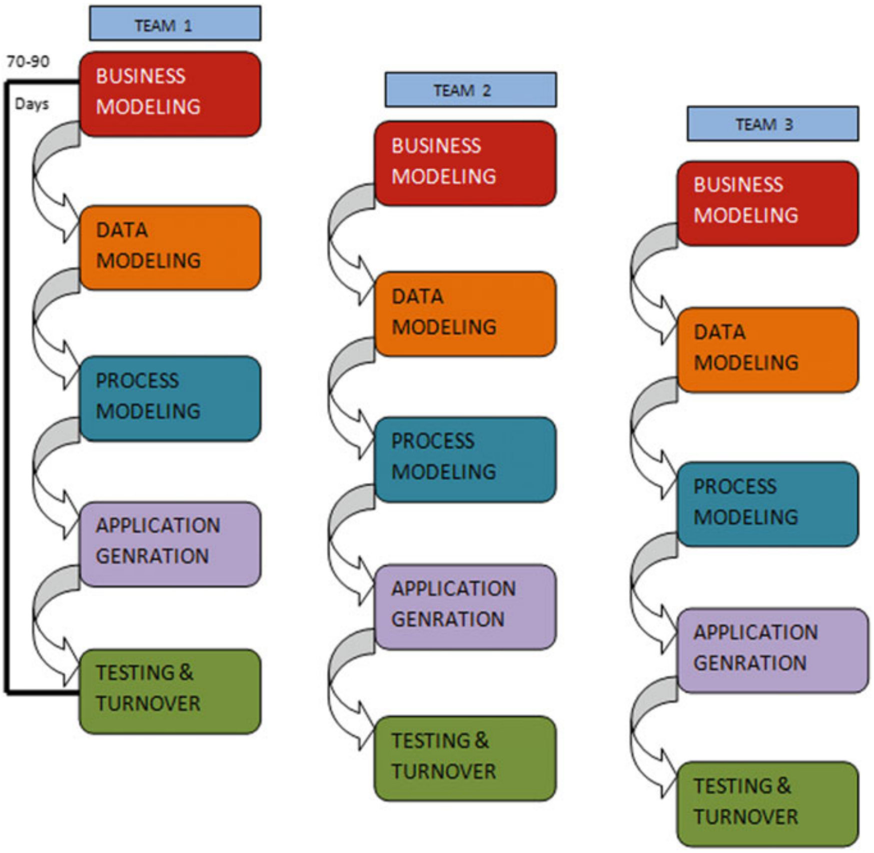


The initial phase of the waterfall is requirement analysis for gathering information about project; find the main purpose or objective of user or customer. Several brainstorming sessions, interviews are conducted during this phase. Next is design phase where all strategies, layouts, and operational details are created. It provides two types of design high-level design and low-level design. The third phase is implementation phase where we write code for design. Then testing phase begins for quality check to find errors or bugs in the product. And then checking whether product's result is according to objective for customer acceptance. Next is deployment phase to check whether test criteria met, whether the product is error free after test and the environment of the product is up to date. Maintenance phase is to fix bugs and problem arise during test phase, make sure the product will work properly, if any change is done write a code, update it, etc.

Incremental Model: It is an updated version of waterfall model. The number of modules created are called build. First of all the requirements are gathered by thorough analysis. Then a first build starts which does design development, testing, implementation. The results of first build are used as input for next build, so further improvement has done on the basis of previous results. All the processing are done incrementally until the product is finished. Incremental model provides flexibility to user. It promotes the maintainability in products. Because each build process small module, it is easy to use and simple to understand. Try to make working model first and analyse their results after that you can build the another models as per needs.

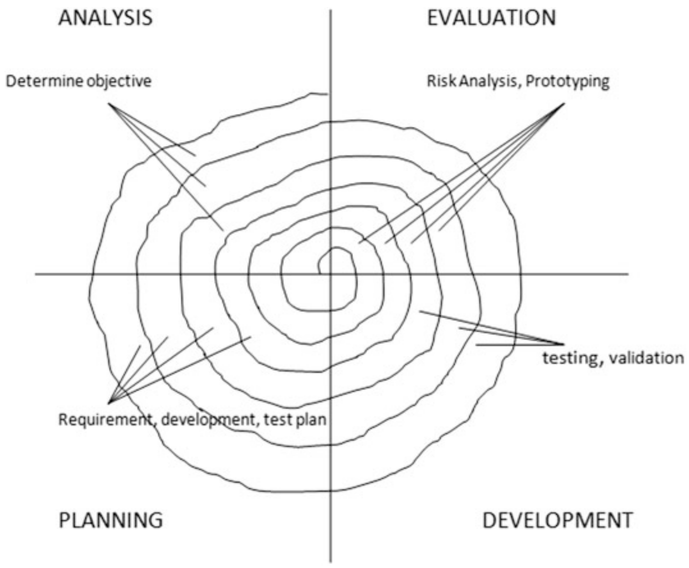


RAD Model: It stands for Rapid Application Development model. It is an advanced version of incremental model. It uses parallel processing which means all the builds are run at same time. Main purpose of the model is to deliver something to see and use. It also provides feedback on each build. There are five phases introduced by the model. The initial phase is business modeling phase which recognizes the data flow between different business methodologies.

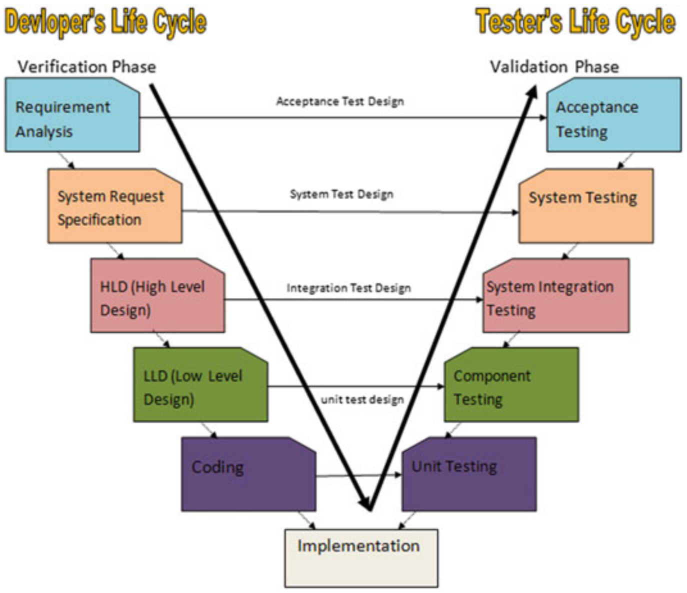


All the facts are gathered and documented during this phase. Next phase is data modeling phase and is used to describe resources according to information collected. Third phase is process modeling phase which uses resources that are described during the second phase and convert them to achieve the objective. Next phase is application generation phase which generates code on the basis of the process model. The last phase is testing phase to find the bugs and check whether the product is according to desire or not for acceptance. The model used for large project and provide more flexibility as compared to other.

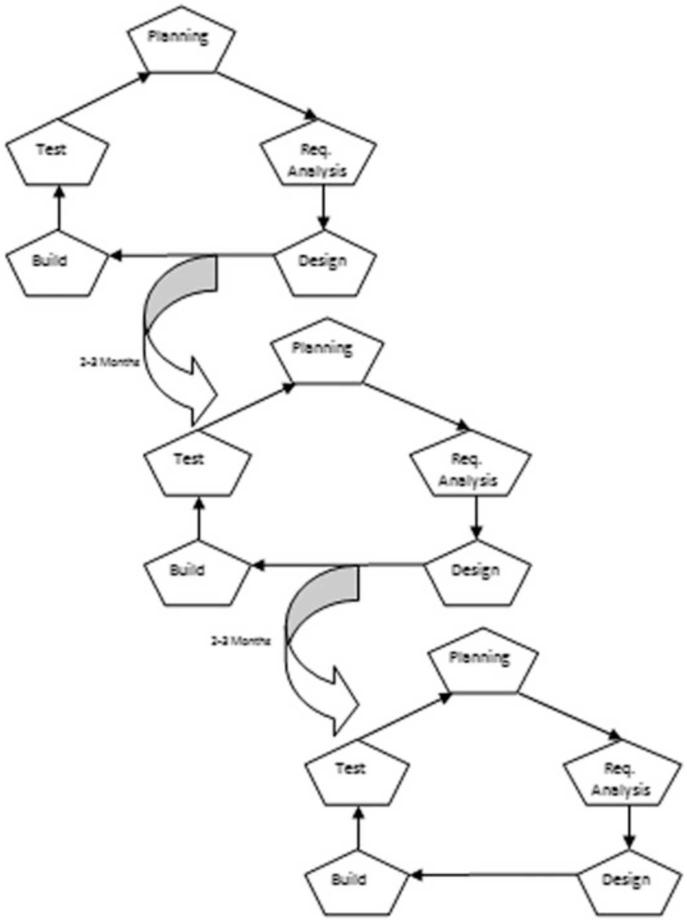
Spiral Model: It is alike to incremental model which is more dominant risk analysis. There are four phases in spiral model: Analysis phase, Evaluation, Development, and Planning. The product passes through each phase repeatedly in iterations. The results of first iteration are used for evaluation so that product is free from anomalies and best changes should be done to get the best product [6].



V Model: It is similar to waterfall model and in addition, every phase is simultaneously tested. Verification and Validation make the V shape of the model. Every stage has its own test case. Once a stage completes its process, its results are validated at the same time. Each phase must be completed before next phase starts [7]. There are five stages in v model and each consists of their own testing criteria.



Agile Model: It is an advanced type of incremental model. Software system is created in progressive, rapid cycle. These results in small progressive deliveries with each delivery built on previous functionality. Agile methodology is more flexible as compared to other. There are some other models that come under agile methodology such as Scrum, Extreme programming (XP), and Rational unified process (RUP) [7]. It is economical, efficient, and provides better quality as compared to the traditional model.



3 Comparison Between Methodologies

As we studied, there are a number of SDLC models. Each has its own specification and working scenario. During the survey, a comparison is done between 4 models on the basis of some characteristics [8], so other remaining models are added to it with more characteristics and a table is made. Most of the models do planning in early stages expect RAD model because it uses small iterations that help to provide results in small time for feedback then further improvements are done on basis of them. Next is requirement specifications that are considered as the first step in each and every model. Further, if we talk about framework type only two models operate in a linear manner, Waterfall and V model, and other models use iterations. The traditional model like waterfall, incremental, V, and spiral model need a big team size. But agile methodology and RAD model can be handled by the small team. Maintenance is a big issue in SDLC models except for RAD model, every other model can provide maintenance up to very little extent. The flexibility of models depends on there working scenarios the waterfall model and v model are working in a linear manner if one stage is complete then we cannot access it again which means they are irreversible. On the other hand, remaining models use iteration which provides too much flexibility to users. Next, the capabilities of the models, only spiral or V model is able to handle large project rest of the model are used for small or medium type projects. Next is detailed documentation, there are three models that is waterfall, spiral and V model which require full documentation and remaining need little of it. Time frame or duration of models depends on there working era. Only RAD and V model complete its processing in short time because they use parallel processing (Table 1).

4 Conclusion

In this paper, analysis of different system development life cycle models has been done by considering various parameters. During the review, we studied several models each has its own working scenario and benefits. Every model minimizes the anomalies came in the previous model. There are two types of model Traditional (waterfall, incremental, RAD, spiral, and V) and Agile (Scrum, RUP, XP). Nowadays, most of the companies use V model because it takes less time to create a software system and parallel testing facilitates better quality that gives a sound product. Large size projects are easily handled by V model and that's why most of the organizations prefer it. An agile terminology helps to handle the medium type of projects and provides more flexibility as compared to the traditional model. Although we have several models but there are some factors that need some improvement, for example, maintenance, improvement in management factors, and need to create a hybrid model by combining existing ones. That should give a better result.

Table 1 Comparison between different SDLC models

S. No.	Properties of model	Water-fall model	Incremental model	Spiral-model	Rad-model	V-model	Agile-model
1.	Planning in early stage	Yes	Yes	Yes	No	Yes	Yes (but little)
2	Returning to an earlier phase	No	Yes	Yes	Yes	No	Yes
3	Handle large project	Not Appropriate	Not appropriate	Appropriate	Not appropriate	Appropriate	Not appropriate
4	Detailed documentation	Necessary or full	Yes but not much	full	Low	Necessary	low
5	Delivery time	Late	Early	Early	Early	Late	Early
6	Cost	Medium	High	Expensive	Low	Low (as compared to waterfall model)	Expensive (as compared to waterfall model)
7	Requirement specifications	Beginning	Beginning	Beginning	Time-boxed release	Beginning	Beginning
8	Flexibility to change	Difficult	Easy	Easy	Easy	Difficult	Easy
9	User involvement	Only at beginning	Intermediate	High	Only at the beginning	Medium	Medium
10	Maintenance	Least	Promotes maintainability	Typical	Easily maintained	Least	Yes provide little bit
11	Duration	Long	Very long	Long	Short	Short	Very long
12	Risk involvement	High	Low	Low	Low	High	High
13	Framework type	Linear	Linear + iterative	Linear + iterative	Linear	V shape	Linear + iterative

(continued)

Table 1 (continued)

S. No.	Properties of model	Water-fall model	Incremental model	Spiral-model	Rad-model	V-model	Agile-model
14	Testing	After completion of coding phase	After every iteration	At the end of the engineering phase	After completion of coding	At each and every phase	In each Iteration
15	Overlapping phases	No	Yes (as parallel development is there)	No	Yes	Yes (as parallel testing is there)	No
16	Re-usability	Least possible	To some extent	To some extent	Yes	Least possible	To some extent
17	Timeframe	Very Long	Long	Long	Short	Short (as compared to waterfall model)	Long
18	Working software availability	At the end of the life cycle	At the end of every iteration	At the end of every iteration	At the end of the life cycle	At the end of the life cycle	At the end of every iteration
19	Objective	High assurance	Rapid development	High assurance	Rapid development	High assurance	Rapid development
20	Team size	Large team	Large team	Large team	Small team	Large team	Small or 10 people
21	Customer control over administrator	Very low	Yes	Yes	Yes	Low	Yes
22	Prototype build	No	Yes	Yes	No	No	No

References

1. Ben-Zahia, Mohamed A., and Ibrahim Jaluta. "Criteria for selecting software development models." *Computer & Information Technology (GSCIT), 2014 Global Summit on*. IEEE, 2014.
2. Velmourougan, S., et al. "Software development Life cycle model to build software applications with usability." *Advances in Computing, Communications and Informatics (ICACCI), 2014 International Conference on*. IEEE, 2014.
3. Velmourougan, S., et al. "Software Development Life Cycle Model to Improve Maintainability of Software Applications." *Advances in Computing and Communications (ICACC), 2014 Fourth International Conference on*. IEEE, 2014.
4. Ragunath, P. K., Velmourougan, S., Davachelvan, P., Kayalvizhi, S., & Ravimohan, R. (2010). Evolving a new model (SDLC Model-2010) for software development life cycle (SDLC). International Journal of Computer Science and Network Security, 10(1), 112–119.
5. Davis A.M, Bersoff E. Hm Comer E.R, A strategy for comparing alternative oftware development life cycle models, IEEE transactions on Software Engineering, Volume: 14, Issue: 10 Publication Year: 1988, Page(s): 1453–1461.
6. S Velmourougan, P Davachelvan, and R Baskaran, Software Reliability Qualification Model, International Journal of Performability. 8(2012) 437–446.
7. Boehm B, "A Spiral of software development and Enhancement", ACM SIGSOFT Software Engineering Notes, ACM, 1986.
8. Thitisathienkul, Patra, and Nakornthip Prompoon. "Quality assessment method for software development process document based on software document characteristics metric." Digital Information Management (ICDIM), 2014 Ninth International Conference on. IEEE, 2014.