

Analysis of Software Development Life Cycle

^[1]Anupam Lakhanpal^[1]Department of Computer Science and Engineering, Galgotias University, Yamuna Expressway Greater Noida, Uttar Pradesh^[1]anupam.lakhanpal@Galgotiasuniversity.edu.in

Abstract: There is a rapid need for better-cost software, further features, quicker distribution, and great-quality software than before it was. The development of the software is becoming increasingly varied and complicated. There are several Software Development Life Cycle models that are extensively used in the development of life cycle. SDLC systems provide a conceptual roadmap to the growth of the software. SDLC model is about reducing the software product's danger and malfunction and optimizing efficiency. The software development method is perhaps the complicated subject without any proper systematic generation method so that the lifespan of the software development methods came into being in order to make it straightforward and standardized. The Software Development Life Cycle specifies the structure that involves various software development phases and functions to be performed. SDLC prototypes are of the greatest importance for the efficient implementation of the software so that it can be produced inside the timeline and should be of the required standard. The paper gives a detailed overview of various SDLC Models its advantages and disadvantages.

Keywords: Prototype Model, SDLC Model, Spiral Model, Waterfall Model.

INTRODUCTION

Nowadays everything is performed through innovation. There is, therefore, a desire to build a further large and complicated software package that meets the increasing and rising demands of the consumer. Software Development Life Cycle is a process through which the software can be continuously produced and which increases the chances of finishing the software project inside the time-limit and retaining the software application quality as per the norm[1]. SDLC, therefore, offers organized and structured coordination of operations to be conducted out in development time to produce good-quality products. SDLC is a technique for data and manufacturing structures planning, creating and maintaining. The Life Cycle Process for System Development offers a series of tasks to be performed by application developers and programmers for

application development. It is often thought of as a part of the life cycle of system design. Any application development method is separated into difficult logical phases that enable an application development organization to effectively coordinate its jobs in order to create a software package with the necessary features within the same specific time period and expenditure[2]. The administration of software development offers the technological way to begin implementing software approaches like coordination, specification implementation, simulation of implementation and layout, system development, evaluation, and assistance. The lifespan of system development includes the range of activities to be done throughout the development of the system and is sometimes called the "life cycle of software development". Software development is categorized into a series of operations that enable easy control of the piece of software by any

application development business. The lifespan of software development begins with the initial examination. In the current system, such a step confirms the issue and acknowledges the need for the solution needed. Initial investigation production determines to choose whether or not to develop the new scheme. The production provides software examination after the initial investigation. The primary purpose of such a process is to determine exactly "what kind of functionality" the program should offer. The Examination of the software includes two kinds of functions

Requirement analysis- All connected information and client specification data are gathered and examined in the requirement analysis.

Software Requirement Specification (SRS) -: SRS is established after client requirements are analyzed. Design proposals are created in SRS which outlines the events to be carried out throughout the program.

Software design occurs in this phase following application analysis; the specifications are provided an ' established ' shape. The programming arrives after developing the process. The programming transforms the meaning of the design into a particular language. Coding process performance provides device validation, in this step; it ensures that the program is mistake-free. Using the Software Development Life Cycle steps the consumer will create the software as needed.

Phases of SLDC

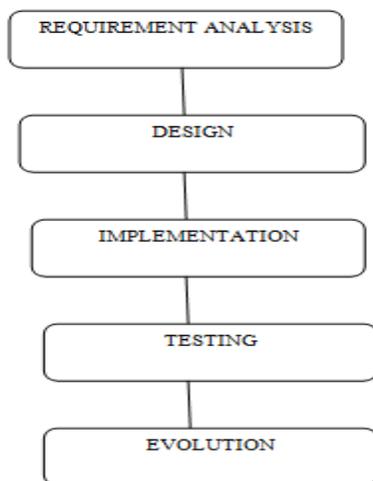


Figure 1: Phase of Software Development Life Cycle

The Phases of Software Development Life Cycle is shown above in Fig. 1 Phases of Software Life Cycle. It is composed of different stages outlining how the software project should be created, planned and managed to ensure that all operational & consumer specifications, requirements and goals are achieved[3]. This assists in manufacturing reliability and service quality. SDLC is a circular technique, repeat stages so that improvements to the layout can be created.

Requirement Analysis- In the requirement analysis process, the customers addresses the software development requirements. The purpose of such a step is to collect all the information of the plan or it can be suggested that the step of the requirement analysis is to collect the information of each specification and to ensure that everyone recognizes the nature of the task and how each specification will be met.

Design- Designers and professional engineers begin the high-level design of the application and process during the design phase in order to allow meeting any specification. The technical aspects of the design are addressed with the investors and different criteria like threats, technology that will be used, group capacity, project limitations, space, and expenditure are checked and afterward, the best design strategy for the product is chosen.

Implementation-This is the process in which all the specifications obtained from the user are applied. In such steps, the coding begins according to the customer's necessity. In this step, each one starts to do the job sys admin starting to make database developers begin coding the feature or we can claim the software components ' and front end programmer statistic create an integrated GUI according to the application specification.

Testing- Testing will be the last stage of the SDLC before clients receive the software. In this step, there is a need to test whether or not the program is functioning according to the standards. It is also possible to verify that the

Software Requirement Specification meets the complete software specification that the customer stated at the period of the settlement.

Deployment and Specification- When software development is finished, it is possible to distribute the software according to customer usage and can provide that normally there is a maintenance team that takes care of any sub-production problems. If a problem is found in the manufacturing, the development team is notified and according on how bad the problem is, it may either involve a warm-fix that is produced and delivered within a short time or, if not quite extreme, it may delay until next edition of the software.

SDLC Models:

Waterfall Model- The waterfall model is the "software process model" which is easiest and conventional. Many users of the waterfall model are for smaller projects. Such model functions in linear sequence. In this design, the outcome of first phase is used as the source to the next phase. Sometimes it's also named "sequential linear model"[4]. The first phase in this design is study of the system. At this stage, study the issue or application needs. Afterward, in the analysis process, the issue and specifications are evaluated, and in the design phase, the specifications are designed and then converted into a programming language is transformed into design framework during coding phase. Coding output is transmitted to the test process. Mistakes in such a step are deleted from the code. The maintenance phase is needed following testing of the entire program. As these steps are carried out one by one and drop like water flying through the air that's why it is named the waterfall model. The Waterfall model is shown below in Fig. 2 Waterfall Model

Advantages:

- It is easy to explain every pattern.
- Explanation is simple.
- Implicit in every step of the research.

Less expensive.

Disadvantages

- Learning to the customer in this model is much less important.
- Problems aren't noticed until system testing.
- Big projects don't help.

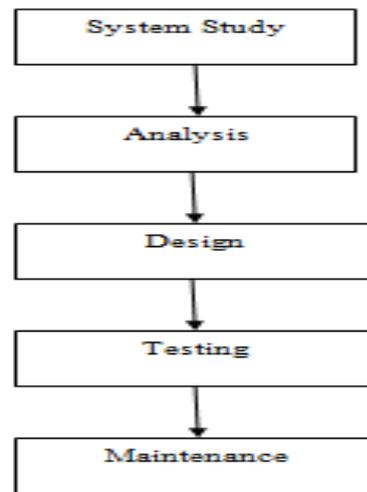


Figure 2: Waterfall Model

Prototype Model

The prototype model is the basis of the process of evolution. Evolutionary process models are the incremental form systems that the user can gradually design more comprehensive app versions using this model[5].

Advantages:

- Easily Accessibility
- Great Results
- Reducible Price
- User will definitely feel the interaction with the process.
- Early Planning
- Disadvantages*
- There is possibility of incomplete project.
- Possibility of malfunctioning system
- No flexibilities.
- Not suited to big systems.

The administration is very difficult. The Prototype model is shown below in Fig. 3 Prototype Model

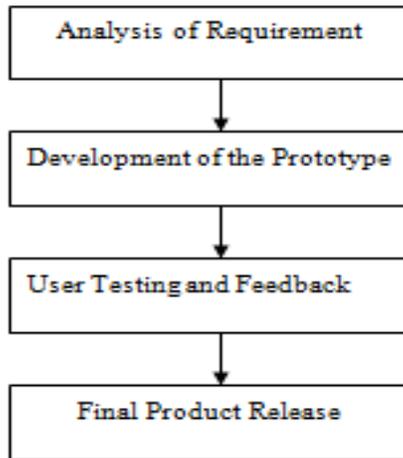


Figure 3: Prototype Model

V Shaped:

The v model is separated into two divisions in which the left section analyzes the software that needs to be created. Testing tasks include the correct section[6]. The model's left and right sections work at the same time. It is not possible to switch to the next phase in this model until or unless it is not possible to complete the preceding step. Because of every phase testing it is not possible to get shielded away from the project objective in this model.

Advantages

- It is simple to understand and execute.
- Removal of bugs in early phase.
- Correct downward mistake circulation.

Disadvantages

- Not versatile and stable design. High risk correlated with such a model.
- The purpose of this model is not explicit.
- It's highly resistant to change.
- More time is expensive and necessary.
- This model does not provide a viable path to the issue discovered during test phases.

The V Model is shown below in Fig. 4 V Model

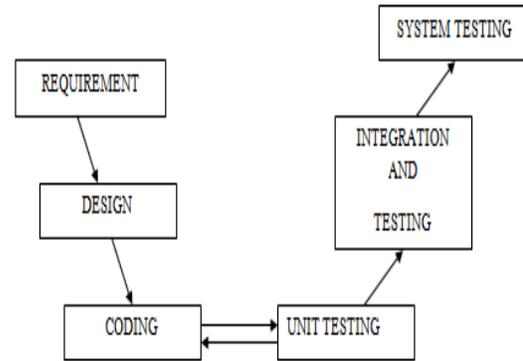


Figure 4: V Model

Spiral Model:

The spiral model is effective for large, onerous and complex projects. The model takes advantage of many of the same steps as the waterfall model[7]. The model consists of many cycles. The process involves four key activities representing four sectors- planning, risk assessment, growth, and customer assessment.

Planning-: Targets are set and strategies are defined in this process.

Risk Assessment-: strategies are assessed established and solved in this context.

Design-: Create, check the product next stage.

Customer Assessment-: Customer assessment of the product in this.

Advantages

- Every moment a different design was obtained.
- Ability to reuse.
- Improved efficiency.

It has a structured approach, step by step.

Errors are removed in early stage.

Disadvantages

- Not a reasonable approximation of the expense and time.
- Risk assessment costs are high for big projects.
- The model is complicated.

RAD (Rapid Application Development) Model

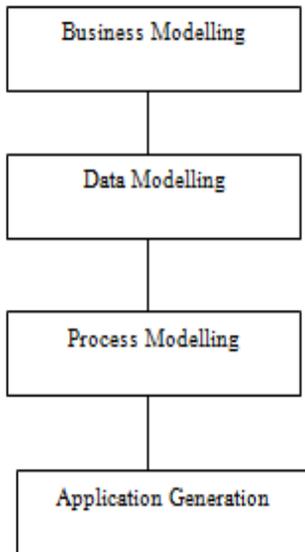


Figure5: RAD Model

RAD model is a platform for the rapid development of applications. It's an iterative kind of model. The elements or functionalities in RAD paradigm are created in connection as though they were mini-projects[8]. The RAD Model is shown above in Fig. 5 RAD Model. The advances are packaged in time, produced and constructed into a working model. The various stages in RAD Model are-

Business Modelling- The flow of data between different corporate tasks is recognized.

Data Modelling- Data collected from enterprise modelling is used to identify the data entities required for the company.

Process Modelling- To accomplish several specific company goal, data elements described in data modelling are transformed to accomplish the company information distribution. Data elements description is recognized.

Application Generation- Computerized systems are used to transform prototypes of processes into software and the system itself.

Advantages

- Time for production is decreased.
- Increases Module Scalability.
- Fast early reviews are taking place.

Encouraging suggestions from customers.

From the very start, automation stops a lot of integration issues.

Disadvantages

It depends on great team results and individual achievement to recognize business needs.

The only system can be developed using RAD that can be modularised.

Highly experienced programmers/designers are required.

It has a strong dependency on modelling expertise.

The expense of data analysis and automatic software creation is extremely unenforceable to smaller schemes.

Comparson of Different Models:

Since there are different SCLC models, each one has benefits and drawbacks based on which design to choose from[9]. For example, if the specifications are identified first and then clearly known and there is a need to have full project management at all times, use the waterfall model. Due to the advent of test strategies during the life cycle, the V-shaped Model has a better chance of succeeding over the waterfall model. This fits well enough for smaller projects where it is easy to understand specifications. The incremental model is at the core of the process of creating cyclic applications. The spiral model is great for big and task-critical projects where significant risk evaluation such as rocket launch is necessary[10]. RAD Design is adaptable to changes because it utilizes fast development cycles i.e. customers immediately get the RAD product. It also requires client involvement thus increasing the probability of embracing the group of potential users and recognizing an overall decline in project threat.

CONCLUSION

There are several Software Development Life Cycle models including, "Waterfall, RAD, Spiral, Incremental, V-shaped, etc" used in different companies based on the current conditions there.

**International Journal of Engineering Research in Computer Science and Engineering
(IJERCSE)**

Vol 4, Issue 11, November 2017

Each of these various software creation frameworks has its own pros and cons. Software Development Life Cycle Model is beneficial for consumers to get a service of superior quality in time and under budget. The Software Development Life Cycle is the compilation of various processes that accompanied for the systematic growth, layout, and upkeep of software projects, and ensures that all customer needs are met with the smallest amount of commodity usage. In this paper we have compared the different software development life cycle models on the basis of certain features like- Requirement specifications, Risk involvement, User involvement, Cost etc. on the basis of these features for a particular software project one can decide which of these SDLC model should be chosen for that particular project. In the technology sector, it is essential to pick the right life cycle model, as the software must be supplied within the deadline and should have the intended consistency.

REFERENCES

1. tutorialspoint, "SDLC - Overview," Tutorialspoint. 2017.
2. J. Broad, "System Development Life Cycle (SDLC)," in Risk Management Framework, 2013.
3. A. Ghahrai, "Software Development Life Cycle – SDLC Phases," Test. Excell., 2017.
4. Yana Sarycheva, "SDLC Waterfall Model," Tutorialspoint, 2019. .
5. Tutorialspoint, "SDLC Software Prototype Model," Tutorialspoint, 2017.
6. V. Osetskiy, "SDLC Models Explained: Agile, Waterfall, V-Shaped, Iterative, Spiral," Existek, 2017. .
7. K. Dowding and B. C. Greene, "Spiral Model," in Encyclopedia of Power, 2012.
8. Model RAD, "SDLC - RAD Model," Tutorialspoint, 2018.
9. A. Mishra and D. Dubey, "A Comparative Study of Different Software Development Life Cycle Models in Different Scenarios," Int. J. Adv. Res. Comput. Sci. Manag. Stud., 2013.
10. S. Malik and C. Nigam, "A Comparative study of Different types of Models in Software Development Life Cycle," Int. Res. J. Eng. Technol., vol. 4, no. 11, pp. 437–440, 2017.
11. Tyagi N., Dhull N., Sharma M., Jain V., Awasthi S. (2020) Coronavirus Spreading Forecasts Based on Susceptible-Infectious-Recovered and Linear Regression Model. In: Hassanien AE., Dey N., Elghamrawy S. (eds) Big Data Analytics and Artificial Intelligence Against COVID-19: Innovation Vision and Approach. Studies in Big Data, vol 78. Springer, Cham. https://doi.org/10.1007/978-3-030-55258-9_1 (October, 2020)
13. Likhesh Kolhe, Yogesh Pingle, Puja Kadam, Dr.Vishal Jain. (2020). Prediction of Coronavirus Covid-19 cases using Linear Regression and Support Vector Machine. International Journal of Advanced Science and Technology, 29(5s), 1911 - 1924. Retrieved from <http://serisc.org/journals/index.php/IJAST/article/view/17901>
14. S.Balamurugan, Dr.P.Visalakshi, V.M.Prabhakaran, S.Charanyaa, S.Sankaranarayanan, "Strategies for Solving the NP-Hard Workflow Scheduling Problems in Cloud Computing Environments", Australian Journal of Basic and Applied Sciences, 8(16): 345-355, 2014

**International Journal of Engineering Research in Computer Science and Engineering
(IJERCSE)**

Vol 4, Issue 11, November 2017
