



# SOFTWARE DEVELOPMENT LIFE CYCLE MODELS: A REVIEW PAPER

**Biswamohan Acharya, Prabhat Kumar Sahu**

Department of Computer Science and Engineering,  
Siksha 'O' Anusandhan (Deemed to be University), Bhubaneswar, Odisha

## ABSTRACT

*Each application's quality is the finish results of how every step of the life cycle of software development has been managed. In order to achieve a good quality product, multiple teams and strategies are used. Software development is a crucial process that everyone is aware of. However, justice will only be done if all the phases are well involved in their respective ways. Different Software Development Life Cycle (SDLC) models are commonly used for the software development. The SDLC models supply a theoretical guide to software development. The SDLC models that are very important for the systematic evolution of the software in such a way that it will be delivered within the time limit & should be of good quality as well. The proper use of SDLC enables project managers to regulate the software's entire development strategy. Every SDLC has its disadvantages and advantages in deciding which model under which situation should be executed. This paper compares various popular life cycle models such as prototype model, waterfall, prototype, rapid development of applications, V-shaped model, spiral model & incremental model.*

**Keywords:** Incremental Model, Spiral Model, Software Development Life Cycle (SDLC), V-Shaped Model, Waterfall Model.

**Cite this Article:** Biswamohan Acharya, Prabhat Kumar Sahu, Software Development Life Cycle Models: A Review Paper, *International Journal of Advanced Research in Engineering and Technology*, 11(12), 2020, pp. 169-176.

<http://iaeme.com/Home/issue/IJARET?Volume=11&Issue=12>

## 1. INTRODUCTION

Everybody knows how important the computer is in our day-to-day life. Today, computers are being used in all areas, such as business, pharmacy, healthcare, schooling, farming, though in the military domain. The activities that can be performed by the computer within a few microseconds, those are very difficult and time consuming for humans. Many companies make the software programs in order to provide the offices with facilities. Some problems occur during early software development [1].

Therefore, some organized flow of software engineering and s/w programs are built to avoid these problems or obstacles. The SDLC is the technique by which the software can be minutely developed & which enlarge the probability of completing the software project within the time limit & maintaining the consistency of the software product as per the standard [2]. The framework of the (SDLC) provides an order of activities to be followed by system designers & developers to develop software.

It is frequently regarded as a subdivision of the life cycle of the system development. Software development process is categorized into different logical steps that allow a software development company to coordinate their work efficiently to create a software by product with all the necessary features within a particular timeframe & budget [3]. All the software projects go through the gathering of requirements, system design, business analysis, implementation, & testing for quality assurance. By using different SDLC model also depends entirely on the developers as the matter of personal option. Every SDLC has its weaknesses & strengths, & in one situation every SDLC can provide better functionality than in another. Then the provocation is to distinct which model under certain circumstances should be choose to provide the specific set of the functionalities. Theoretically, one life cycle models may suited in particular conditions & at the common time another model may look fitting in with the requirements, but when deciding which model to choose it should be considered trade-off [4].

## 2. SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)

These are the models that help develop the software developers are looking for. It's the software life cycle's detailed and diagrammatic visualization. It includes all the activities necessary to move a software product through its phases of the life cycle. In other words, from its foundation to retrial, it plans the variety of activities performed on a software product [5]. Fig. 1 shows the multifarious SDLC phase.



**Figure 1** Illustrates various Software Development Life Cycle (SDLC) phases.

**1. Requirements:**

Requirement is one of the most important steps of recognizing the client's need. For the consistency of the criteria, there will be several review meetings. All findings of the review should be logged and tracked. They suggest that both formal & informal interviews should be held with the applicant's right stakeholders. This will help developer to get a true view of what the application is expected to do. Clearly document these discovery, so the recline of the group is well aware of the need. Consequently, it helps to reduce the defects developed by the requirements alone.

**2. Design:**

Specifications are transform to use the case diagrams & comprehensive business-related designs documentation provided.

**3. Development:**

This Phase carried out by the development group in which the structure documents are inputs together with the update of the technical reviews. Each code must be placed under the team's scanner, such as having an inspection through the code developed & the test cases of the unit must also be reviewed prior to execution.

**4. Testing:**

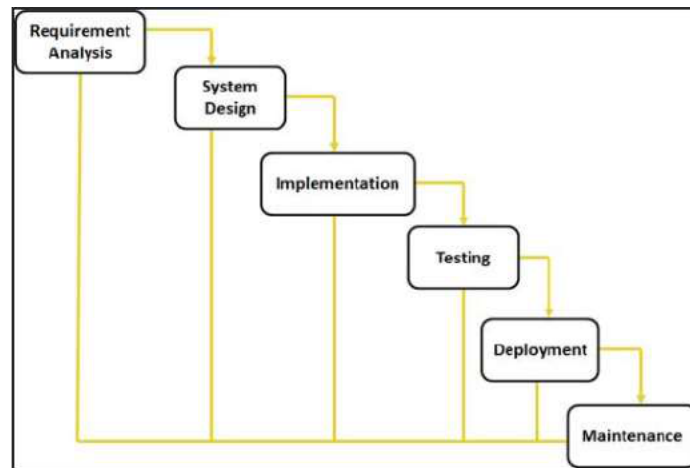
The SDLC key validation stages is the testing phase. The focus on the complete testing of the applications developed on the basis of the matrix of requirements.

**5. Maintenance:**

A technical analysis meeting should be held to analyze & finalize the maintenance phase in order to structure the results and issues being considered [6].

### **3. WATERFALL MODEL**

Royce suggested the waterfall model, also known as linear sequential life cycle software development model. The Linear Sequential Model is the software engineering classic model. This model is one of the oldest & is frequently used in the government projects & in many firms. Due to this model accentuate in early-stages of planning, it prevents pre-development design flaws. Moreover, its detailed documentation and preparation make it work well for projects where a major concern is quality control [7].



**Figure 2** Depicts the waterfall models.

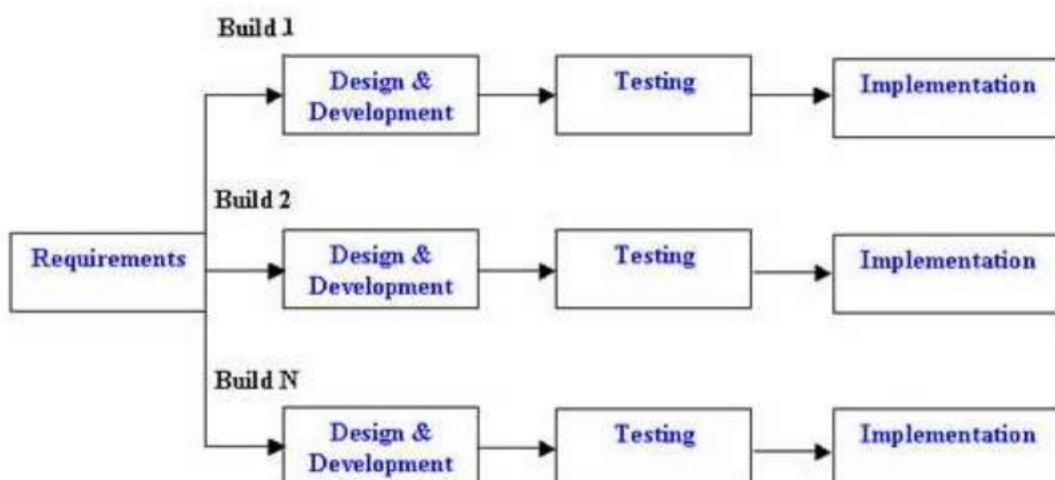
### 1. Waterfall Model Problem Area:

The waterfall model is illustrated above in Fig. 2. This section describes the major threats which are given below in the waterfall model.

- If the problem occurs at the early phase of a model, then the changes in that phase are very difficult to do.
- Deliverable product is not manufactured in a short period of time.
- High risks & uncertainty.
- This model is not used for complicated, object-oriented projects.
- Not ideal for projects with frequent changes in specifications.

## 4. INCREMENTAL MODEL

The incremental model combines linear sequential model elements (repetitively applied) with prototyping iterative philosophy. The Incremental model petition linear sequences as calendar time progresses in a phased fashion. In the incremental model, such as prototyping and other evolutionary approaches, is iterative in design. Unlike prototyping, however, with each step, the incremental model focuses on producing an actual product [8].



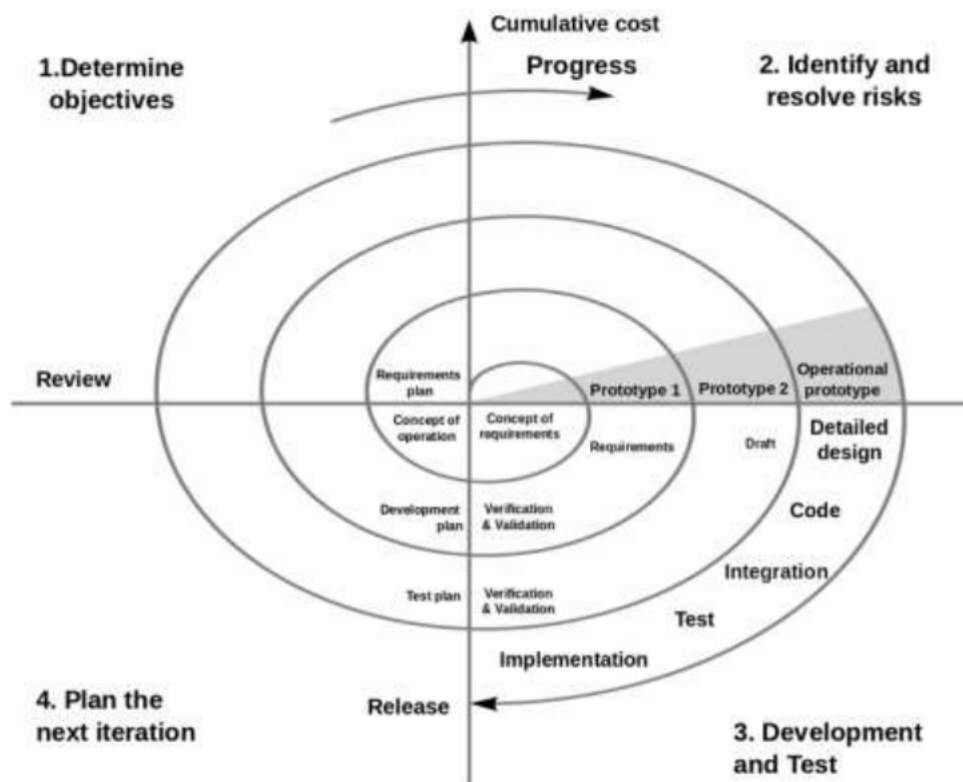
**Figure 3** Illustrates the incremental model.

*Incremental Model Problem Area:* The incremental model major issues are described below. Fig. 3 depicts the block diagram for the incremental model.

- It requires high-quality planning and design.
- Clear interpretation of the entire system is required before it must be shattered down & progressively developed.
- The total cost of the incremental model is higher than linear sequential model.

## 5. SPIRAL MODEL

This is the evolutionary software operation model incorporating the continual nature of the prototyping with the regulated & systematic features of the linear sequential model. The spiral model using, software is created in an order of gradual release. The incremental releases may be a paper models or prototype during the early iterations. More and more absolute variety of the engineered systems are produced during later iterations [9].



**Figure 4** Illustrates the spiral model [8].

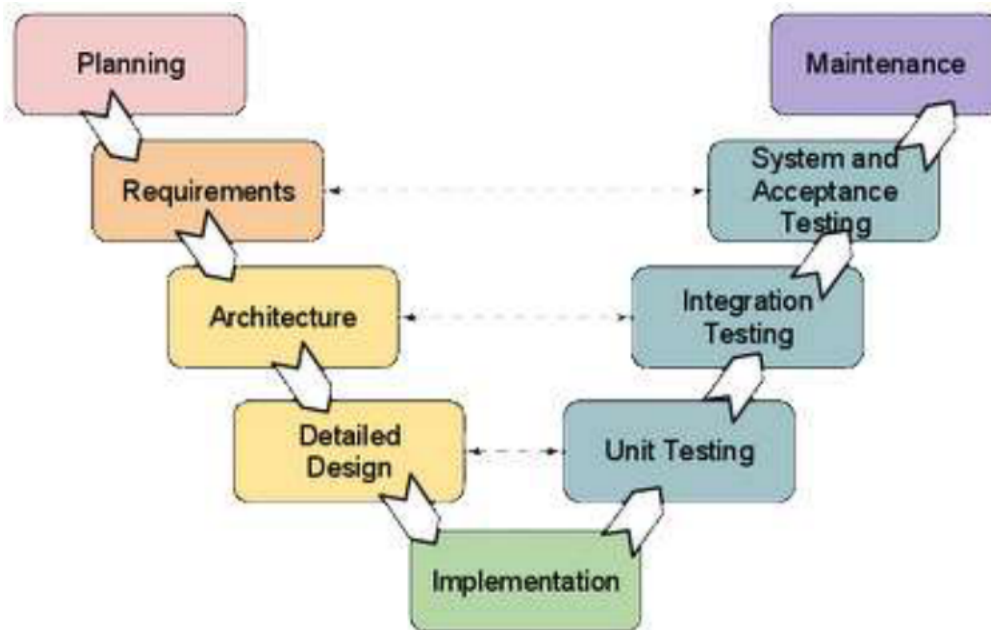
### 1. Spiral Model Problem Area:

The spiral model major problems are described in this section. Fig. 4 illustrates the spiral model.

- This model is very expensive.
- To carry out the highly specific, risk analysis expertise is required.
- Project achievement is highly dependent on the phase of risk analysis.
- For smaller projects, this model does not work well.

## 6. V-SHAPE MODEL

The V-Shape model is a model for validation & verification. The V-Shaped model life cycle is the sequential process execution path, like the waterfall models. Before the next stage begins, every phase has to be completed. The product testing is planned in parallel with a corresponding development phase [10].



**Figure 5** Illustrates the block diagram of the V-shape Model.

### 1. *V-Shape Model Problem Area:*

This section describes the major issues in the V-Shaped Model. Fig. 5 shows the block diagram of the V-shape Model. The Table 1 shows the comparative analysis of the SDLC models.

- Quite rigid & less flexible.
- During the implementation phase, software is developed so that no early software prototypes will be produced.
- If any changes occur at midway, the test documents must be updated along with the required documents.
- For small to medium-sized projects where requirements are clearly defined and fixed, the V-shaped model should be used.

**Table 1** Comparison of SDLC Models [7].

FEATURES	WATERFALL	V-SHAPED	INCREMENTAL	SPIRAL
Requirement specifications	Beginning	Beginning	Beginning	Beginning
Cost	Low	Expensive	Low	Expensive
Simplicity	Simple	Intermediate	Intermediate	Intermediate
Risk involvement	high	Low	Easily manageable	Low
Expertise	High	Medium	High	High
Flexibility to change	Difficult	Difficult	Easy	Easy
User involvement	Only at beginning	At the beginning	Intermediate	High
Flexibility	Rigid	Little flexible	Less flexible	flexible
Maintenance	Least	Least	Promotes maintainability	Typical
Duration	Long	According to project size	Very long	Long

## 7. CONCLUSION

Most SDLC models such as waterfall, spiral, incremental V-shaped etc. are used in different organizations depending on the conditions. All these different models of software development have their own pros and cons. The hybrid of all these methods is used in the software industry, with some modification. Since researching all models through the various factors, it was found that the linear sequential model is used in database-related projects in which the project team members are aware of all the specifications before the project starts. With new technology, incremental model is used and where resources are not available. Prototype model used for development of online transaction processing systems and web interfaces, i.e. where there is a very high level of end-user interaction. Spiral model is used for high-risk, long-term, and low-cost projects where modifications are expected.

## REFERENCES

- [1] V. Rastogi, "Software Development Life Cycle Models- Comparison, Consequences," Int. J. Comput. Sci. Inf. Technol., 2015.
- [2] R. Arora and N. Arora, "Analysis of SDLC Models," Int. J. Curr. Eng. Technol., 2016.
- [3] S. S., "A Study of Software Development Life Cycle Process Models," SSRN Electron. J., 2017, doi: 10.2139/ssrn.2988291.
- [4] S. S. Ghuman, "SDLC Models- A Survey," Int. J. Comput. Sci. Mob. Comput., 2013.
- [5] A. R. Otero and A. R. Otero, "System Development Life Cycle," in Information Technology Control and Audit, 2018.
- [6] M. Geogy and A. Dharani, "Prominence of each phase in Software development life cycle contributes to the overall quality of a product," 2015, doi: 10.1109/ICSNS.2015.7292390.

- [7] S. S. Kute and S. Thorat, "A Review on Various Software Development Life Cycle (SDLC) Models," Int. J. Res. Comput. Commun. Technol., 2014.
- [8] 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.
- [9] P. Trivedi and A. Sharma, "A comparative study between iterative waterfall and incremental software development life cycle model for optimizing the resources using computer simulation," 2013.
- [10] V. Osetskyi, "SDLC Models Explained: Agile, Waterfall, V-Shaped, Iterative, Spiral," Existek, 2017.