



# SOFTWARE DEVELOPMENT MODELS AND THEIR REAL ASPECTS

**Divya Tanwar**

*Sbs , Vivekananda Institute of Professional Studies, GGSIPU, Delhi*

## **ABSTRACT**

*Software development life cycle models are the most integral part when it comes to software development which can fully justify today's digitally environments with increasing mobile, web, and desktop applications. This research deals with a vital and important issue in information technology with respect to various models used in development. It is concerned with the software management processes that examine the area of software development through the development models, which are known as software development life cycle. It represents five of the development models namely, waterfall, Iteration, prototype and spiral and their real time examples which we can relate in our day to day life. These models are complex and difficult to identify related to real time scenario.*

*Therefore, the main objective of this research is to represent different models of software development and make a comparison between them to show the features and defects of each model. This paper demystifies these concepts so that the SDLC is correctly understood with their real time.*

**Keywords:** *Iterative Prototyping SDLC, Software development life cycle, Spiral model, Real aspects, Waterfall model*

## **I. INTRODUCTION**

Software development life cycle models are the important part of the Software Engineering which is a discipline whose aim to develop quality software, which is on time, with minimum cost and can satisfy the organization needs. There are basically four main models in SDLC with different characteristics and drawbacks [2] .The most difficult task is to identify the model which suits the organization structure, for this every organization need to know the model and its real time application so the user can relate, understand and find out how its actually working. Every model in SDLC is the combination of various steps, which step leads to what n what is the basic requirement of that step is need to be understood.

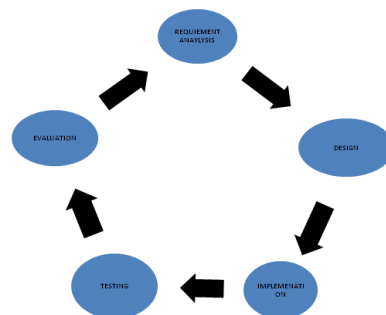
## **II. WHAT IS SDLC (SOFTWARE/SYSTEM DEVELOPMENT LIFE CYCLE)**

It is a well defined process by which a system is planned, developed and implemented [1]. The system development starts with the requirement for improving their business system.

There are following activities involves in SDLC

- Preliminary Investigation (Problem Identification)
- Feasibility study
- System analysis
- System designing
- Development of software
- System testing
- Implementation & Evaluation
- Maintenance

} Requirement Analysis



**Fig 1: Software/System Development Life Cycle**

1. Preliminary Investigation OR Problem Identification:

One of most difficult task of the system analyst is identifying the real problem of the existing system. It defines the user requirements or what the user expects from the new system. This also includes the rough idea of the resource requirements as well as estimated time for completion and number of persons expected to be involve in each phase [1].

Problem identification helps in :

- Defining a problem
- Setting proper system goal
- Determining the boundaries of the project by considering the limitations of available resources

2. Feasibility study:

It determines the possibility of either improving the existing system or developing the complete new system. The purpose of feasibility study is to determine whether the requested system successfully realizable.

There are four aspects of feasibility study :-

- Technical feasibility
- Economical feasibility
- Operational feasibility
- Behavioural feasibility

3. System analysis:

It involves detailed understanding of all important facts of the business area under investigation. This requires data collection from a verity of sources such as questionnaires, forms, interviews, study of existing documents. It can be involved the direct observation in the organization and collected documents to understand the whole



#### 4. System designing:

In this process the primary object is to identify user requirements and to build a system that satisfies these requirements. Design of the system is mainly the logical design that can be sketch on a paper or on a computer. It includes physical design elements, describes the data to be inputted.

The process involved in manipulation of data & output design represents:-

- File structure, storage devices etc
- Database is also designed in this phase
- Changes to be made in the organisational structure of the firm are outlines
- Input, Output, files, forms and procedures are planned
- Finally standards for testing, documentation, system control are designed.

#### 5. Development of software:

Development is a phase where detailed design is used to actually construct and build the system. In this phase the system is decided whether to buy commercial software or to develop new customized program with the help of the programmers. The choice depends upon the cost of software and cost of programming.

#### 6. System testing:

Testing is a process of making sure that the program performs the intended task. Once the system is designed it should be tested for validity. During this phase the system is used experimentally to ensure that software does not fail and it will work according to its specification. It is tested with special test data.

#### 7. Implementation & Evaluation:

This is the final phase of development. It consists of installing hardware, programs, collecting data and organizing people to interact with and run the system. In this phase user actually starts using the system therefore it also involves training of users and provides friendly documentation.

Evaluation is the process of verifying the capability of a system after it put into operation to see whether it meets the objective or not. It includes response time, overall reliability and limitations user behavior.

#### 8. Maintenance:

It is process of incorporating changes in the implemented existing system.

- Enhancement: Adding new functions or additional capability of the system.
- Adaptation: Customizing the software to run in a new environment.
- Correction: Correcting the bugs in the existing software.

### III. SDLC MODELS AND THEIR REAL ASPECTS

#### 3.1 SDLC Waterfall Model

The Waterfall Model was first development Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very easy to recognize and use.

“In a waterfall model, each phase must be concluded (over and done) before the next phase can start and there is no overlapping in the phases[2].”

Waterfall model is the most basic SDLC approach that was used for software development .

## 3.1.1 Waterfall Model design

In "The Waterfall" approach, the entire process of software development is separated into separate phases. In Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

Following is a diagrammatic representation of different phases of waterfall model.

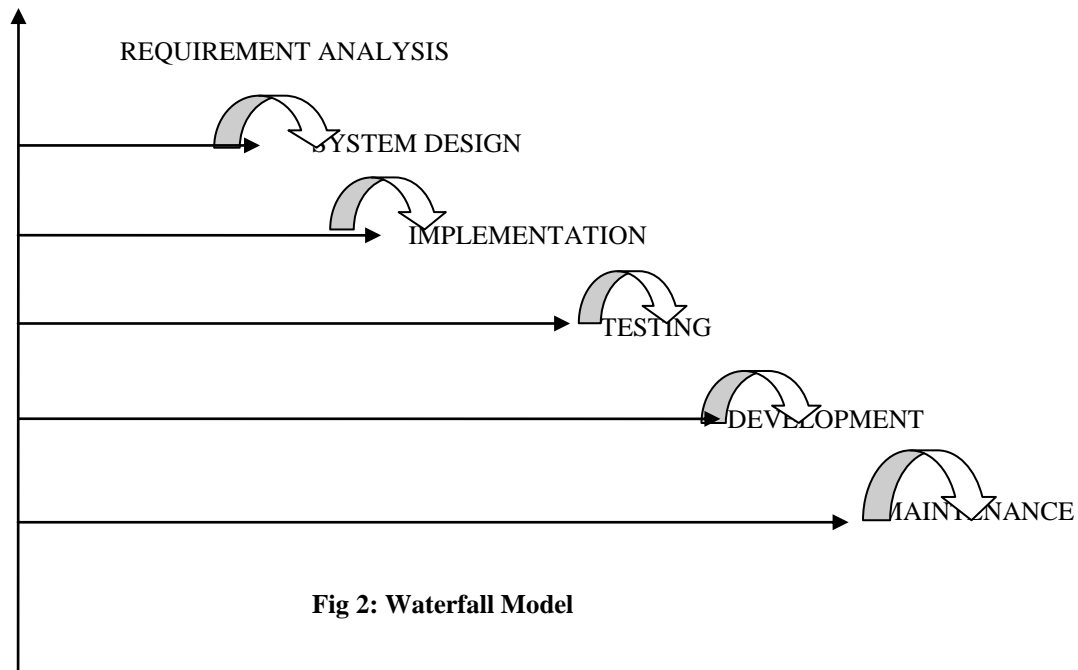


Fig 2: Waterfall Model

The sequential phases in Waterfall model are:

- Requirement Gathering and analysis: All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc known as RSR document.
- System Design: The requirement specifications from first phase are studied in this phase and system design is prepared which can be unit or in various phases. Here we develop the overall system architecture while specifying hardware and system requirements of the system.
- Implementation: With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.
- Integration and Testing: All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- Deployment of system: Once the functional and non functional testing is done, the product is deployed in the customer setting or released into the market.
- Maintenance: There are some issues which come up in the client environment. To fix those issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model phases do not overlap[2].



### 3.1.2 Waterfall Model Application

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are:

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

### 3.1.3 Real time Application: Waterfall model is like a making of TEA where all the above specified phase is same like

- Requirement Gathering and analysis: Find out the need of making Tea
- System Design: Making of tea in various phases, following the sequential order.
- Implementation: Here the execution is done, finally how we are making tea which can be in units like boiling of water, putting sugar and tea and then milk separately
- Integration and Testing: Here all the ingredients are put together which are in above phase are in units and the finally testing of tea is done
- Deployment of system: Here the tea is served to the specific customers
- Maintenance: Tea is preserved, for the further uses.

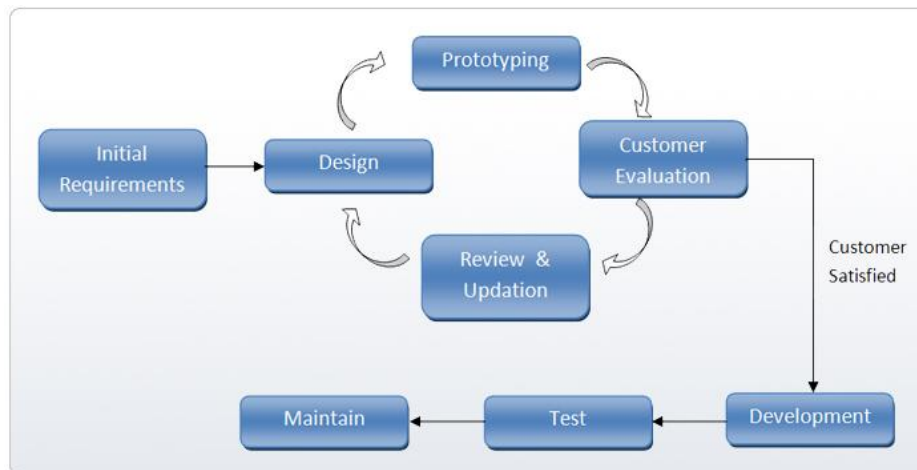
Waterfall model is same as making of tea as once the process of making tea is done cannot be revert back.

## 3.2 Prototyping Model

Prototyping approach, also known as evolutionary approach, came to picture because of failures that occurred in the final version of the software application developed using the waterfall approach. The failure generally occurs because of the changes in the requirement of the proposed system or because of the gap in understanding the user requirement by the development team[9]. A gap in the first version of the developed application, unavoidably leads to the need for redoing the application. To overcome these limitations, the concept of prototyping was introduced [8].

### Prototyping Approach

A prototype is the sample implementation of the system that shows limited and main functional capabilities of the proposed system. After a prototype is built, it is delivered to the user for the evaluation. The prototype helps the user determine how the feature will function in the final software. The customer provides suggestion and improvements on the prototype[11].



**Fig 3: Prototype model**

The development team implements the suggestion in the new prototype, which is again evaluated by the user. The process continues until the user and the development team understands the exact requirement of the proposed system [7]. When the final prototype is developed, the requirement is considered to be frozen.

The prototyping approach is used in the requirement gathering and in the analysis phase to capture the exact requirement of the proposed system. After the requirements are frozen, the remaining phases of the development process needs to be executed to complete the development of the software system [10].

### 3.2.1 Need for a prototype in software development

There are several uses of a prototype. An important purpose is to illustrate the input data formats, messages, reports, and the interactive dialogues to the customer. This is a valuable mechanism for gaining better understanding of the customer's needs:

- How the screens might look like
- How the user interface would behave
- How the system would produce outputs

This is something similar to what the architectural designers of a building do; they show a prototype of the building to their customer. The customer can evaluate whether he likes it or not and the changes that he would need in the actual product. A similar thing happens in the case of a software product and its prototyping model.

Another reason for developing a prototype is that it is impossible to get the perfect product in the first attempt. Many researchers and engineers advocate that if you want to develop a good product you must plan to throw away the first version. The experience gained in developing the prototype can be used to develop the final product.

### 3.2.2 Real time Application: Prototype model is like a making of an E-COMMERCE WEBSITE where all the above specified phase is same like

1. An e-commerce website, such as shopping site is an example where you can implement the prototyping approach.
2. You can develop the prototype of the various web pages of the shopping site such as catalogue page, product order page etc., and present it to the customer for approval.
3. If the customer approves the prototype of the site, requirements are states again and the design of the web site is initiated.

4. If the customer does not approve the web site, the development team revisits the prototype and resubmits it to the customer for approval.

5. This process continues until the prototype.

### 3.3 Incremental/Iterative Model

Iterative design is an approach of incrementally developing and refining a design based on feedback and evaluation. Iterative design can apply to a learning experience, the creation of media, or the development of learning systems [3].

Some practical examples of iterative design:

- Wiki - A wiki is a natural repository for iterative design. The 'Page History' facility allows tracking back to prior versions. Modifications are mostly incremental, and leave substantial parts of the text unchanged.
- Common law - The principle of legal precedent builds on past experience. This makes law a form of iterative design where there should be a clear audit trail of the development of legal thought.
- Evolution - There is a parallel between iterative and the theory of Natural Selection. Both involve a trial and error process in which the most suitable design advances to the next generation, while less suitable designs perish by the wayside. Subsequent versions of a product should also get progressively better as its producers learn what works and what doesn't in a process of refinement and continuous improvement [6].

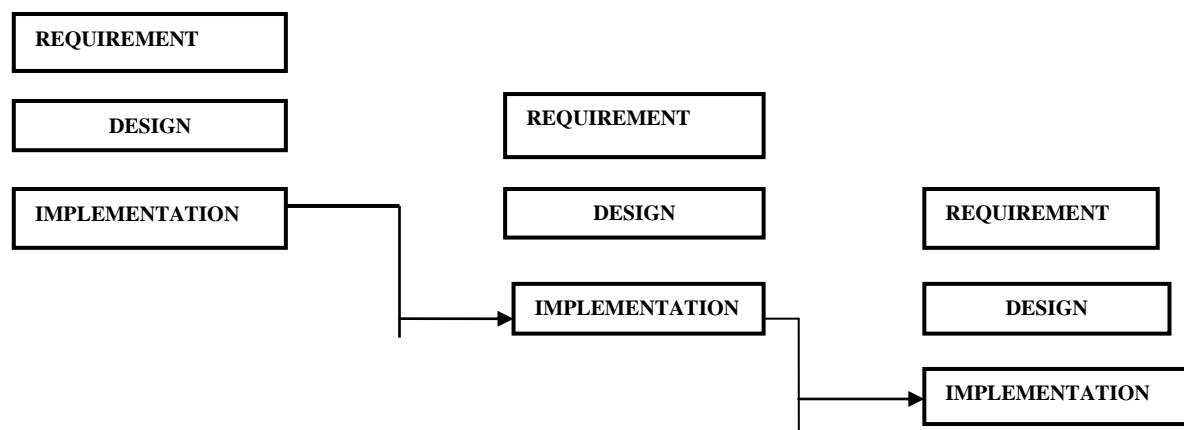


Fig 4: Prototype Model

#### 3.3.1 Real time Application: Repeated tasks which need to be modified at every step are the example of Iterative model

- A musician may start with one version of a song and refine it, making changes to a note here and there or changing the tempo and speed. Many different iterations of a song may be created this way until the optimal finished product has been created that sounds and works the best.
- A cook may experiment with a recipe, tweaking the ingredients or changing different steps of the process slightly until the food tastes as good as it possibly can.
- Microsoft has released multiple versions of the Windows operating system to add functionality for users and to correct bugs. Each different version is a different iteration of its operating system and the goal is for the next iteration to be better than the last.

## 3.4 Spiral model

The spiral model is called a meta model since it encompasses all other life cycle models. Risk handling is inherently built into this model. The Spiral model appears like a spiral with many loops. The exact number of loops in the spiral is not fixed. Each loop of the spiral represents a phase of the software process. Spiral Model includes the iterative nature of the prototyping model and the linear nature of the waterfall model. This approach is ideal for developing software that is revealed in various version[5] [6]. In each iteration of the spiral approach, software development process follows the phase-wise linear approach. At the end of first iteration, the customer evaluates the software and provides the feedback. Based on the feedback, software development process enters into the next iteration and subsequently follows the linear approach to implement the feedback suggested by the customer. The process of iteration continues throughout the life of the software.

### 3.4.1 Spiral Approach Phases

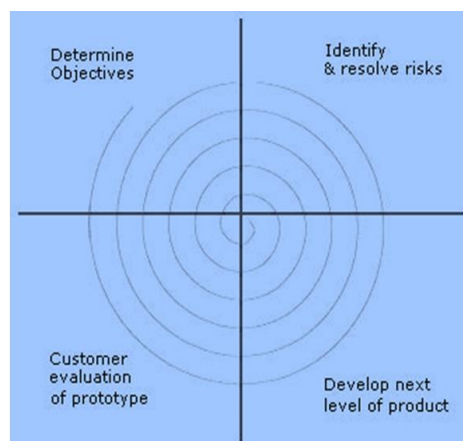
1. Customer Communication: Includes understanding the system requirements by continuous communication between the customer and the system analyst.
2. Planning: Includes estimating Schedule, cost, and resource for the iteration.
3. Risk Analysis: includes identifying, estimating, and monitoring technical and management risks, such as schedule slippage and cost overrun.
4. Engineering: Includes requirement gathering and design of the software system.
5. Construction and release: Includes coding, testing and deploying software at the customer site and providing user-support documents.
6. Customer Evaluation: Includes evaluation of software by the customer and implementing feedback in the next iteration of the software development.

#### a) First quadrant (Objective Setting)

- During the first quadrant, it is needed to identify the objectives of the phase.
- Examine the risks associated with these objectives.

#### b) Second Quadrant (Risk Assessment and Reduction)

- A detailed analysis is carried out for each identified project risk.
- Steps are taken to reduce the risks. For example, if there is a risk that the requirements are inappropriate, a prototype system may be developed.



**Fig. 5: Spiral Model**



## c) Third Quadrant (Development and Validation)

- Develop and validate the next level of the product after resolving the identified risks.

## d) Fourth Quadrant (Review and Planning)

- Review the results achieved so far with the customer and plan the next iteration around the spiral.
- Progressively more complete version of the software gets built with each iteration around the spiral.

### 3.4.2 Real time Application of spiral model :

Working on the missiles or satellites is the real time example of a spiral model. The risk associated and cumulative costs both are the very important aspect of the missiles.

1. Understanding the basic requirements, need and goal is the first step of the spiral model.
2. Estimate the schedule, time, cost and the various resources to make the components or the part of a missile is comes under the category of planning which is the second iteration of spiral model .
3. Identifying, estimating, and monitoring various risks associated covered under risk association.
4. Developing the final model based upon the approved requirements is the developing iteration of the spiral model which is called Engineering which Includes requirement gathering and design of the software system.
5. Includes coding and testing and launch of a missile is the most crucial iteration of spiral model which is called Construction and release

Last iteration is the Evaluation, feedback or the limitation or the success story of launching of missile

## IV. CONCLUSION

SDLC Models play a very significant role in our organization. The most important aspect is to understand their real time application with their core features as the original water fall model is used by various big companies for their internal projects where their specification is well defined. Iterative water fall model overcome the drawback of original waterfall model. It allows feedback to proceeding stage. Prototype model used to develop online systems for transaction processing. Since significantly reduce rework and lead to the creation of working model in lower capital cost. Spiral model is used for development of large, complicated and expensive projects like scientific Projects .Since spiral model approach enables the project term to address the highest risk at the lowest total cost.

## REFERENCES

- [1] Software Development Life Cycle (SDLC) – the five common principles.htm
- [2] Software Methodologies Advantages & disadvantages of various SDLC models.mht
- [3] Craig Larman, Victor R. Basili, "Iterative and Incremental Development: A Brief History," Computer, vol. 36, no. 6, pp. 47-56, June 2003, doi:10.1109/MC.2003.1204375.
- [4] Curtis, Krasner, Iscoe, 1988.
- [5] Dr. Winston W. Royce (1929 - 1995) at www.informatik.uni-bremen.de. Retrieved 27 Oct 2008.
- [6] Conrad Weisert, Information Disciplines, Inc., Chicago, 8 February, 2003
- [7] Craig Larman, Victor R. Basili (June 2003). "Iterative and Incremental Development: A Brief History.
- [8] Ian Sommerville, Software Engineering, Addison Wesley, 9th ed., 2010.

- [9] Jim Hurst, "Comparing Software Development Life Cycles," SANNS Software Security, 2014.
- [10] Sanjana Taya and Shaveta Gupta, "Comparative Analysis of Software Development Life Cycle Models," IJCST Vol. 2, Issue 4, Oct . - Dec. 2011
- [11] Boehm B, "A Spiral Model of Software Development and Enhancement", ACM SIGSOFT Software Engineering Notes", "ACM", 11(4)