

# A Comparison of Waterfall, Spiral, Agile Models

Geetika Sethi

Asst. Prof. A.S. College, Khanna

**Abstract:** - This paper deals with the Software Management processes that examine the area of software development through the development models which are known as software development lifecycle model. This paper includes three SDLC Models, namely: Waterfall Lifecycle Model, Spiral model, Agile model. The main objective of this research is to make a comparison between models to show the features and limitation of each model.

**Keywords:** — Software Development Lifecycle, Software Management Processes, Comparison between three models

## I. INTRODUCTION

The various activities that are undertaken when developing software are commonly modeled as a software development life Cycle (SDLC). The software development life cycle begins with the identification of requirements for software and ends with the retirement of the software. The software development lifecycle is a program of high level model choice is the first step in the project planning process. There are a no. of different models for software development lifecycle ultimately, the choice of lifecycle model determines the organization of software development activities over time, effecting when software will be delivered, how often, and with what resources. The term software lifecycle is not even agreed upon, with software process being at least one alternatives terms. However, most software development life cycles are phased processes with clearly identifiable goals, milestones and tasks. There exist no general agreement on which phases should be included in a software lifecycle or the order in which they occur. In this paper the comparison between different types of models are done which are:

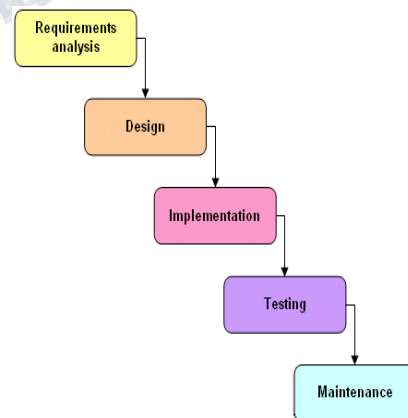
1. Waterfall model
2. Boehm's Spiral Lifecycle Model
3. Agile Model

1. Waterfall Lifecycle Model: the waterfall Lifecycle Model was introduced by Winston Royce in 1970 and is currently the most commonly used model for software development. The idea of waterfall model is :There are different stages to the development and the output of the first stage flow into the second stage and these outputs flow into third stage and so on.

There are two variations of waterfall Lifecycle Model:

- (a)Original Waterfall Lifecycle Model
- (b)Modified Waterfall Lifecycle Model

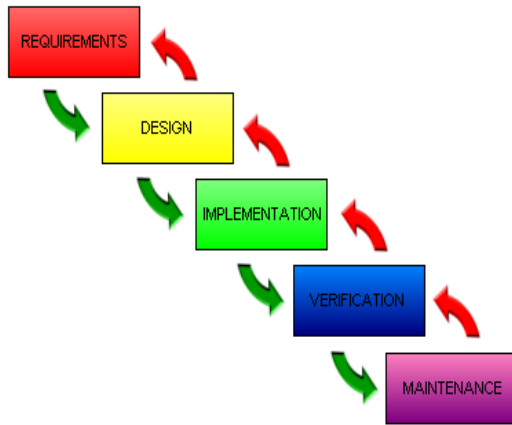
**(a) Original Waterfall Lifecycle Model:** it emphasizes that software is developed in sequential phases with established milestones, documents and reviews the end of each phase. This model is also known as traditional software Lifecycle Model. In this model requirements are frozen before start the process. This model can be called as “one way street, with no turning back”, where once analysis is complete, design begun, analysis cannot be reentered.



**Figure (1): Original Waterfall Lifecycle Model**

**(b) Modified Waterfall Lifecycle Model:** the modified waterfall lifecycle model represents an experience based refinement of the original waterfall Lifecycle model. This model was introduced to overcome the limitations of the original model. The model needs to provide for overlap

and feedback between phases. The following are the major extensions in this model over the original model. It introduces iteration between the phases along with the restriction of providing iteration if possible, only between successive phases in order to reduce the expense of revision that results from iteration over multiple phases. The provision for verification and validation of the phase output in the software Lifecycle are added. Needed to provide for overlap and feedback between phases.

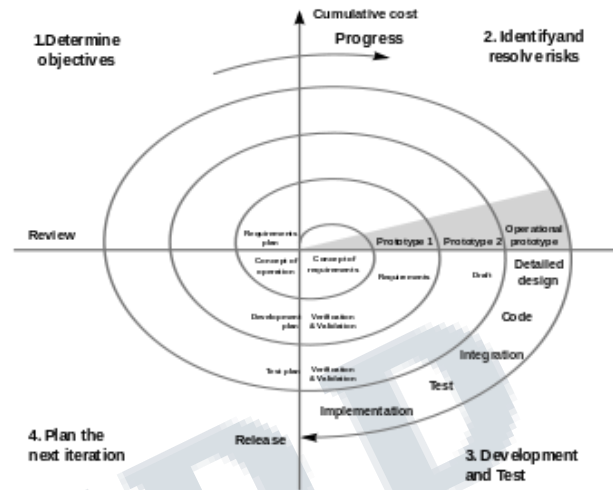


**Figure (2): Modified Waterfall Lifecycle Model**

2. Boehm's Spiral Lifecycle Model: The spiral Lifecycle model combines elements of the waterfall Lifecycle Model, along with an emphasis on the use of risk Management Technique. Boehm proposed a spiral model where each round of the spiral identifies:

The sub problem which has the highest risk associated with it and finds a solution for that problem. Each cycle of the spiral uses the same high-level processes that are outlined below:

- (a) Determine objectives, alternatives and constraints.
- (b) Evaluate alternatives, identify and resolve risks
- (c) Develop, verify next-level product & plan next Phase.



**Figure (3): Spiral Model**

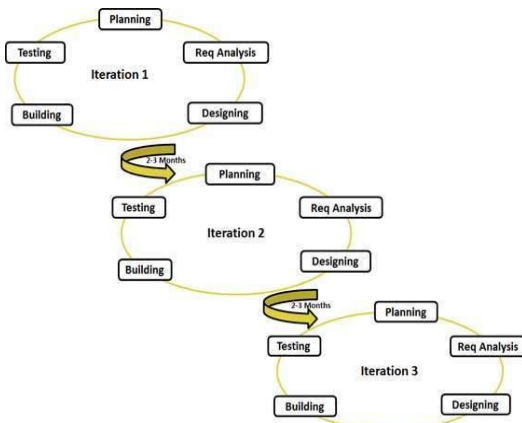
**Perform four basic activities in every cycle**

This invariant identifies the four basic activities that must occur in each cycle of the spiral model:

- (i) Consider the win conditions of all success-critical stakeholders.
- (ii) Identify and evaluate alternative approaches for satisfying the win conditions.
- (iii) Identify and resolve risks that stem from the selected approach(es).
- (iv) Obtain approval from all success-critical stakeholders, plus commitment to pursue the next cycle.

3. Agile Model: Agile Model believes that every project needs to be handled differently and the existing methods need to be alter according to the project requirements. The aim of agile model is to reduce overheads and to be able to respond quickly to changing requirements without excessive rework. Agile thought process had started early in the software development and started becoming popular with time due to its flexibility and adaptability. Agile model used where a software company is developing a small or medium sized product. In agile model the tasks are divided to time frames or small time frames to deliver specific features for a release.

Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features, the final build holds all the features which are required by the customers. There is a commitment from the customer to become involved in the development process.



**Figure (4): Agile Model**

**Pros**

1. It is vary realistic approach to software development.
2. After each iteration delivers early partial working solutions.
3. Easy to manage.
4. Flexible model.
5. Suitable for both types of requirements fixed and changing.

**Cons:**

1. Not suitable for complex projects.
2. More risk of sustainability and maintainability
3. Depends heavily on customer interaction
4. Lack of documentation

**Methodology**

**Method for Selecting the Model for the Project**

1. Nature of Project
2. Project size
3. Project duration
4. Project complexity
5. Level and type of expected risk
6. Level of understanding of user requirement
7. Level of understanding of the application area
8. Customer Involvement
9. Experience of developers
10. Team size
11. Man-machine interaction
12. Availability of tools and technology
13. Version of the product
14. Level of consistency require

**Results**

**Comparison of Models**

Features	Original water fall	Modified water fall
Requirement Specification	Beginning	Beginning
Understanding Requirements	Well Understood	Not Well understood
Cost	Low	Low
Availability of reusable component	No	Yes
Complexity of system	Simple	Simple
Risk Analysis	Only at beginning	No Risk Analysis
User Involvement in all phases of SDLC	Only at beginning	Intermediate
Guarantee of Success	Less	High
Overlapping Phases	No overlapping	No Overlapping
Implementation time	long	Less
Flexibility	Rigid	Less Flexible

**TABLE [1]-Comparison between Original Waterfall Model and Modified Waterfall Model**

Features	Spiral model	Agile model
Requirement Specification	Beginning	Till Ending
Understanding Requirements	Well Understood	Well Understood
Cost	Expensive	Expensive
Availability of reusable component	Yes	Yes
Complexity of system	Complex	Complex
Risk Analysis	Yes	No
User Involvement in all phases of SDLC	High	High
Guarantee of Success	High	High
Overlapping Phases	Yes Overlapping	No Overlapping
Implementation time	Depends on project	Depends on customer requirement
Flexibility	Flexible	Flexible

**TABLE[2]-Comparison between Spiral and Agile Model**

### CONCLUSION

After completing this research, it is concluded that modified waterfall model overcome the drawback of original waterfall model. It allows feedback to proceeding stage. Spiral model is used for development of large, complicated and expensive projects like scientific projects. Agile model is used where user requirements continuously changing. In this model user interact every phase of SDLC.

### FUTURE WORK

- ❖ Some topics can be suggested for future work:
- ❖ Suggesting a model to simulate advantages that are found in different models to software process management.
- ❖ Apply the suggested model to many projects to ensure of its suitability and documentation to explain its mechanical work.

### REFERENCES

- [1]. Vanshika Rastogi “ Software Development Lifecycle Model Comparison, Consequences” International Journal of Computer Science and Information Technologies, Vol. 6(1), 2015, 168-172.
- [2]. Pressman : Software Engineering, Tata-McGraw Hill Publishing House.
- [3]. [http://www.tutorialspoint.com/sdlc/sdlc\\_agile\\_model.htm](http://www.tutorialspoint.com/sdlc/sdlc_agile_model.htm).
- [4]. Shikha Maheshwari and Prof Dinesh ch.Jain “A comparative Analysis of Different Types of Model in Software Development Lifecycle” International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 5, May 2012
- [5]. Rajib Mall: Fundamentals of Software Engineering, PHI Learning Pvt. Limited.
- [6]. Nabil Mohammed Ali Munassar and A. Govardhan “ A Comparison betn Five Models of Software Engineering” IJCSI, Vol.7, Issue 5, September 2010.
- [7]. Jalota, Pankaj: An integrated approach to Software Engineering, Narosa Publishing Company.
- [8]. Gill, Nasib Singh: Software Engineering: Software reliability, Testing and Quality Assurance. Khanna Book Publishing.
- [9]. <http://www.cs.cscr.edu/-stan/classes/cs530/-slides/ch3.pdf>