



## A Comparative Analysis of Different types of Models in Software Development Life Cycle

Ms. Shikha maheshwari<sup>1</sup>

Research Scholar in Dept. of C.S.E.

S.V.I.T.S, Indore(M.P)

shikhamaheshwari0906@gmail.com

Prof.Dinesh Ch. Jain<sup>2</sup>

Reader, S.V.I.T.S,Indore

dineshwebsys@gmail.com

**Abstract-** In the present scenario all software systems are imperfect because they cannot be built with mathematical or physical certainty, Hence in this research paper the comparison of various software development models has been carried out. According SDLC each and every model have the advantage and drawbacks so in this research we have to calculate the performance of each model on behalf of some important features. The concept of system lifecycle models came into existence that emphasized on the need to follow some structured approach towards building new or improved system. Many models were suggested like waterfall, Iterative model, prototype model, spiral model etc.

**Keywords-** Software Development Process, SDLC, phase of SDLC models, Comparative analysis of model, four models.

### I. INTRODUCTION

A software development process, also known as a software development life cycle (SDLC), is a structure imposed on the development of a software product. It is often considered as a subset of system development life cycle. There are several models for such processes, each describing approaches to a variety of activities that take place during the process.

Software Engineering (SE) is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, and the study of these approaches; that is, the application of engineering to software because it integrates significant mathematics, computer science and practices whose origins are in Engineering. Various processes and methodologies have been developed over the last few decades to improve software quality, with varying degrees of success. However, it is widely agreed that no single approach that will prevent project overruns and failures in all cases. Software projects that are large, complicated, poorly-specified, and involve unfamiliar aspects, are still particularly vulnerable to large, unanticipated problems.

A software development process is a structure imposed on the development of a software product. There are several models for such processes, each describing approaches to a variety of tasks or activities that take place during the process. It aims to be the standard that defines all the tasks required for developing and maintaining software. Software Engineering processes are composed of many activities, notably the following:

- Requirements Analysis ,
- Specification
- Software architecture
- Implementation
- Testing
- Documentation
- Training and Support
- Maintenance

Software development teams, taking into account its goals and the scale of a particular project, and have a number of well-established software development models to choose from. Therefore, even though there are number of models each software Development Company adopts the best-suited model, which facilitates the software development process and boosts the productivity of its team members.

There are four types of Model are:

- 1 Waterfall
- 2 Iterative
3. Prototype
4. Spiral.

### II. PHASES OF SDLC

Problem solving in software consists of these activities:

- Understanding the problem
- Deciding a plan for a solution
- Coding the planned solution
- Testing the actual program

For large systems, each activity can be extremely complex and methodologies and procedures are needed to perform it efficiently and correctly. Furthermore, each of

the basic activities itself may be so large that it cannot be handled in single step and must be broken into smaller steps. For example, design of a large software system is always broken into multiple, distinct design phases, starting from a very high level design specifying only the components in the system to a detailed design where the logic of the components is specified. The basic activities or phases to be performed for developing software system are

- Determination of System's Requirements
- Design of system
- Development (coding) of software
- System Testing

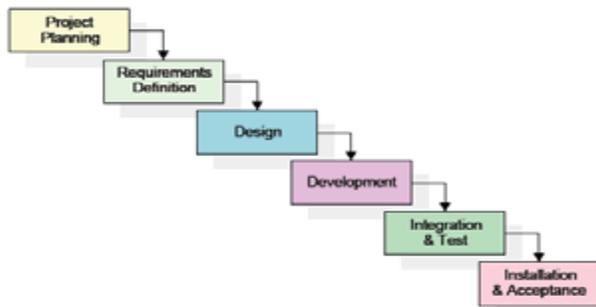


Fig1. Software Development Life Cycle

### III. SOFTWARE DEVELOPMENT MODELS

#### A) Original Waterfall Model

The Waterfall model is a conventional, linear, sequential or traditional waterfall software life cycle model. It is a sequential development approach, in which development is seen as flowing steadily downwards through the phases of requirements analysis, design, implementation, testing (validation), integration, and maintenance.

##### 1) Basic Principles

- Project is divided into sequential phases, with some overlap and splash back acceptable between phases.
- Emphasis is on planning, time schedules, target dates, budgets and implementation of an entire system at one time.
- Tight control is maintained over the life of the project via extensive written documentation, formal reviews, and approval/signoff by the user and information technology management occurring at the end of most phases before beginning the next phase.

##### 2) Advantage of original water model:

- Each stage has well defined deliverable or milestone.
- It is simple to use and understand.

- It acts as template into which methods for analysis, design, code test and maintenance can be placed.

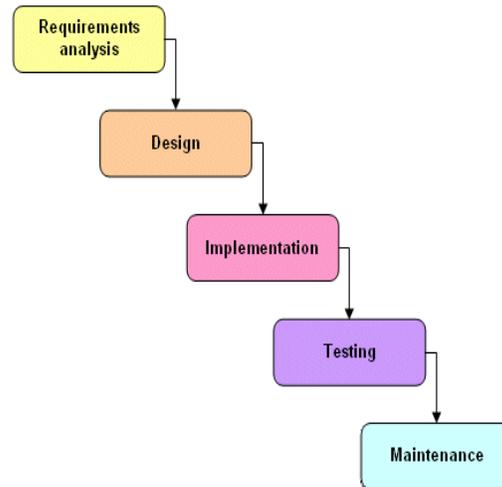


Fig2. Original Waterfall Model

##### 3) Disadvantage of original water model:

- The biggest disadvantage of the waterfall model is one of its greatest advantages. You cannot go back a step; if the design phase has gone wrong, things can get very complicated in the implementation phase.
- Often, the client is not very clear of what he exactly wants from the software. Any changes that he mentions in between may cause a lot of confusion.
- Small changes or errors that arise in the completed software may cause a lot of problems.
- Another major disadvantage of the waterfall model is that, until the final stage of the development cycle is complete, a working model of the software does not lie in the hands of the client.

#### B) Iterative Waterfall Model

##### 1) Basic Principles

- The problems with the Waterfall Model created a demand for a new method of developing systems which could provide faster results, require less up-front information and offer greater flexibility.
- Iterative model, the project is divided into small parts. This allows the development team to demonstrate results earlier on in the process and obtain valuable feedback from system users.

- each iteration is actually a mini-Waterfall process with the feedback from one phase providing vital Information for the design of the next phase.

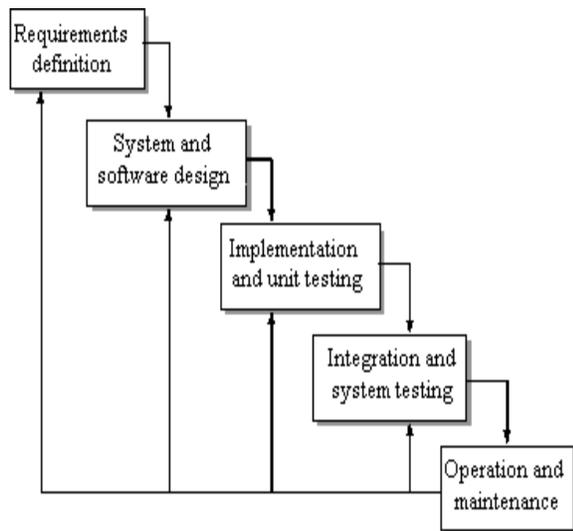


Fig3. Iterative waterfall model

2) Advantages-

- It is much better model of the software process.
- It allows feedback to proceeding stages.
- It can be used for project wherein the requirements are not well understood.

2) Disadvantages-

- It is not easy to manage this model.
- No clear milestones in the development process.
- No stage is really finished.

C) Prototyping Model

Software prototyping is the development approach of activities during software development, the creation of prototypes, i.e., incomplete versions of the software program being developed.

1) Basic Principles

- Not a standalone, complete development methodology, but rather an approach to handling selected parts of a larger, more traditional development methodology.
- Attempts to reduce inherent project risk by breaking a project into smaller segments and providing more ease-of-change during the development process.
- User is involved throughout the development process, which increases the likelihood of user acceptance of the final implementation.
- Small-scale mock-ups of the system are developed following an iterative modification process until the Prototype evolves to meet the users requirement .
- While most prototypes are developed with the expectation that they will be discarded, it is

possible in some cases to evolve from prototype to working system.

- A basic understanding of the fundamental business problem is necessary to avoid solving the wrong problem.

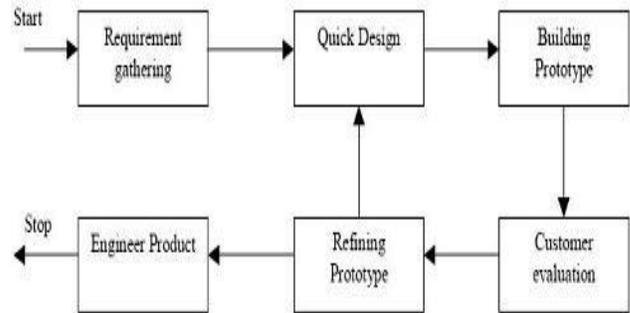


Fig4. Prototype Model

2) Advantages-

- Early visibility of the prototype gives users an idea of what the final system looks like Encourages active participation among users and producer.
- Enables a higher output for user.
- Cost effective (Development costs reduced)
- Increases system development speed
- Assists to identify any problems with the efficacy of earlier design, requirements analysis and coding activities.

3) Disadvantages-

- Possibility of causing systems to be left unfinished.
- Possibility of implementing systems before they are ready.
- Producer might produce a system inadequate for overall organization needs.
- Often lack flexibility.
- Not suitable for large applications.
- Project management difficulties.

D) Spiral Model

The spiral model is a software development process combining elements of both design and prototyping-in-stages, in an effort to combine advantages of top-down and bottom up concepts. It is a meta-model, a model that can be used by other models.

1) Basic Principles

- Focus is on risk assessment and on minimizing project risk by breaking a project into smaller segments and providing more ease-of-change during the development process, as well as providing the opportunity to evaluate risks and weigh consideration of project continuation throughout the life cycle.
- Each cycle involves a progression through the same sequence of steps, for each part of the

product and for each of its levels of elaboration, from an overall concept-of-operation document down to the coding of each individual program.

- Each trip around the spiral traverses four basic quadrants: (1) determine objectives, alternatives, and constraints of the iteration; (2) evaluate alternatives; Identify and resolve risks; (3) develop and verify deliverables from the iteration; and (4) plan the next iteration.
- Begin each cycle with an identification of stakeholders and their win conditions, and end each cycle with review and commitment.

- Every time a new prototype is obtained ,it is reevaluated again and again by the customer .so more customer involvement is there.
- Better productivity through reuse capabilities.
- Proper control over cost ,time and manpower requirement for a project work.
- Errors are eliminated in early phases of project development only.

3) *Disadvantages-*

- This model requires risk identification, its projection, risk assessment and risk management which is not an easy task.
- Cost and time estimations are also not very easy.
- This model is not suitable for smaller project as then the cost of risk analysis is greater than cost of the entire project.

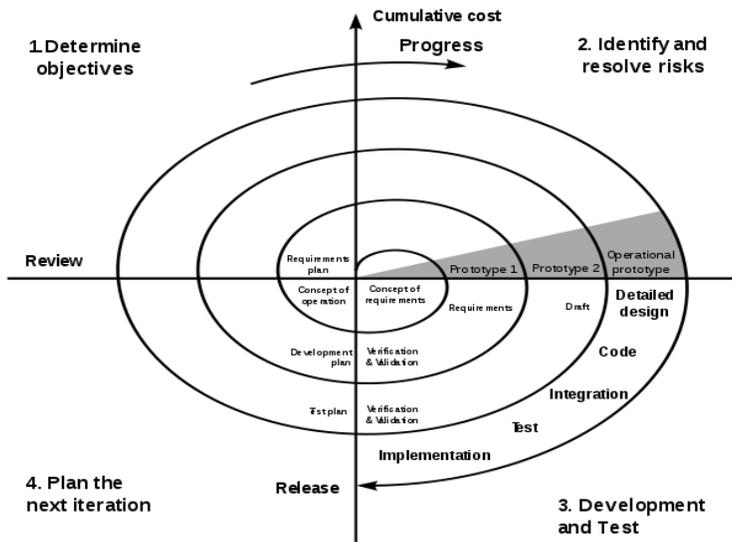


Fig 5 Spiral model

IV. Comparative Analysis of Four Models

Features	Original water fall	Iterative water fall	Prototyping model	Spiral model
Requirement Specification	Beginning	Beginning	Frequently Changed	Beginning
Understanding Requirements	Well Understood	Not Well understood	Not Well understood	Well Understood
Cost	Low	Low	High	Expensive
Availability of reuseable component	No	Yes	yes	yes
Complexity of system	Simple	simple	complex	complex
Risk Analysis	Only at beginning	No Risk Analysis	No Risk Analysis	yes
User Involvement in all phases of SDLC	Only at beginning	Intermediate	High	High
Guarantee of Success	Less	High	Good	High
Overlapping Phases	No overlapping	No Overlapping	Yes Overlapping	Yes Overlapping
Implementation time	long	Less	Less	Depends on project
Flexibility	Rigid	Less Flexible	Highly Flexible	Flexible

Changes Incorporated	Difficult	Easy	Easy	Easy
Expertise Required	High	High	Medium	High
Cost Control	Yes	No	No	Yes
Resource Control	Yes	Yes	No	Yes

Table1 Comparison of four models.

## V. CONCLUSION

After analysis of all models through the various factors, it has been found that the original water fall model is used by various big companies for their internal projects .Since the development team is familiar to the environment and it is feasible to specify all requirements of working environment. Iterative water fall model overcome the drawback of original waterfall model.It allow 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.

## ACKNOWLEDGMENT

Shikha Maheshwari pursuing Master Of Engineering in Computer science branch from SVITS indore. She has done research work in the area of software engineering.

I would like to thanks to Prof. Dinesh Chandra Jain for supporting in this research paper.

## REFERENCE

- [1]Dr.Deepshikha Jamwal"Analysis of software Development Models", ISSN: 2229-4333(print),ISSN:09768491(online), vol.IISSUE 2, Decemder 2010.
- [2]Chan, D.K.C. Leung, K.R.P.H, "Software development as a workflow process", 2-5 Page(s): 282 – 291, Dec. 1997.
- [3]Kushwaha ety.al, "Software Development Process and Associated Metrics - A Framework", IEEE CNF, Page(s):255 – 260, July 2006.
- [4]Rothi, J.,Yen, D, "System Analysis and Design in End User Developed Applications", Journal of Information Systems Education, 1989
- [5]Boehm, B. W. "A Spiral Model of Software Development and Enhancement", ISSN: 0018-9162, Volume: 21, Issue: 5, on page(s): 61-72, May 1988.
- [6]Swapanja Ingale "Comparative Study Of Software Development Model" International conference on advances in computing &management 2012
- [7]Jovanovich, D., Dogsa, T. "Comparison of software development models," Proceedings of the 7th International Conference on, 11-13 June 2003, ConTEL 2003, pp. 587-592.
- [8]Maglyas, A.; Nikula, U.; Smolander, K.,"Comparison of two models of success prediction in software development projects", Software Engineering Conference (CEE-SECR), 2010 6th Central and Eastern European on 13-15 Oct. 2010, pp. 43-49.
- [9] A. M. Davis, H. Bersoff, E. R. Comer, "A Strategy for Comparing Alternative Software Development Life Cycle Models", Journal IEEE Transactions on Software Engineering ,Vol. 14, Issue 10, 1988
- [10]Sanjana Taya "Comparative Analysis of Software Development Life" ISSN:2229-4333(print),ISSN:0976-8491(online),vol.2ISSUE 4,Oct-Dec 2011
- [11]Molokken-Ostvoid et.al, "A comparison of software project overruns - flexible versus sequential development models", Volume 31, Issue 9, Page(s): 754 – 766, IEEE CNF, Sept.2005.
- [12] S. M. Metev and V. P. Veiko, Laser Assisted Microtechnology, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998

