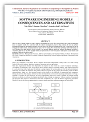
A comprative study of sdlc model

Mr. Vivek Bhatnagar



International Journal of Application or Innovation in Engineering & Management (IJAIEM)
Web Site: www.ijaiem.org Email: editor@ijaiem.org
Volume 4, Issue 10, October 2015 **ISSN 2319 - 4847**

A COMPARATIVE STUDY OF SOFTWARE DEVELOPMENT LIFE CYCLE MODELS

Manzoor Ahmad Rather[1] , Mr. Vivek Bhatnagar[2]

[1]Research Scholar MMICT&BM Maharishi Markandeshwar University Mullana (Ambala)

[2] Associate Professor, MMICT&BM Maharishi Markandeshwar University Mullana (Ambala)

ABSTRACT

Software project development is one of the most powerful, vital and important issue in the world of computers. It is the subset of system development life cycle (SDLC). SDLC is all about the minimization of risk and failure and maximization of quality of the software product. The software development process is the very complex thing without any proper step by step generating procedure so to make the software development processes simple and systematic the software development life cycle came in to existence. This is the systematic and structural method of software developing process. The SDLC defines the framework that includes different activities and tasks to be carried out during the software development process. There are various software development life cycle models that are used in the software development process heaving their own advantages and disadvantages in this paper we have included five of these SDLC models like waterfall model, v shaped model, prototype model, spiral model and iterative model for comparison of existing models.

Keywords: Software Development Life Cycle, Models, Prototype, Modeling, Development, Risk Analysis and Comparative Analysis.

1. INTRODUCTION

The software project planning management is the clear need in now a day. A plan is like a road map. Software engineering is a discipline for developing high quality software for computer based systems. Software engineering is all about the quality focus, process, methods and tools that are used in software development process. The software development process defines a framework that includes different activities and tasks. We can say The process defines what different activities and tasks are to be carried out during software development.(or method of developing the software). The software planning management provides the technical way starting how to implement the software methods that include Communication, Requirement Analysis, Analysis and Design modeling, Program construction, Testing and support. Software development life cycle SDLC is the systematic approach to complete the software development process within the time and maintain quality of the software. System development life cycle provides the set of activities to be carried out during the system development and it is often called that software development life cycle. Software development is divided into set of activities that allow any software development company to control the software product easily. The software development life cycle models use the step by step approach to complete the software development process. If the process is strong, the end product will also be strong and project can get success. While developing the good software product the developers that are directly or indirectly included in this process should keep the following points in their mind.

- A quality focus.
- Process
- Methods
- Tools.

The software process model is the representation of process it presents the description of a process as.

- Specification.
- Design.
- Validation
- Evolution.

The software development life cycle is all about.

- Understanding the problem. i.e. (problem domain).
- Decide a plan for solution. i.e. (solution domain).
- Coding the planed solution.
- Test the actual program.
- Maintain the product.

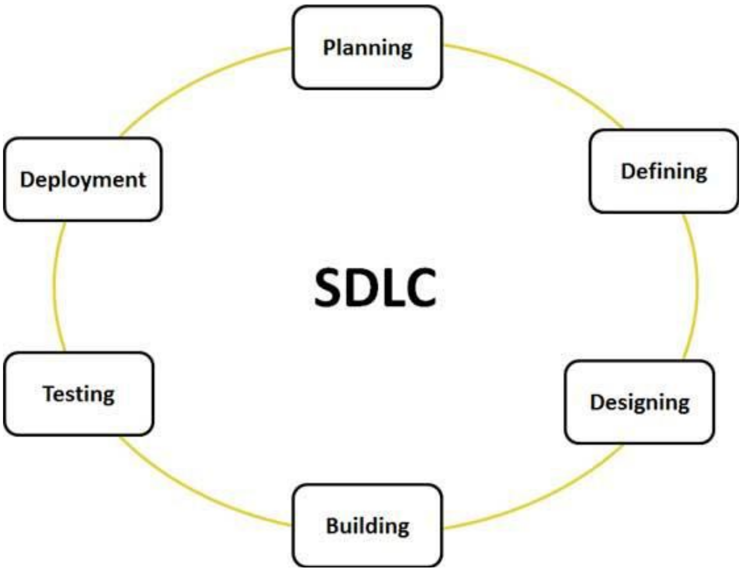


Figure 1 SDLC

The software development life cycle is all about to meet the business needs of the customer.

THE SOFTWARE DEVELOPMENT LIFE CYCLE MODELS.

2. The Waterfall Model

This model was proposed by Royce in 1970. This is the classical model of the software engineering. The model is one of the oldest models used in the government projects and in many of the major companies. The model is also called as the linear sequential model or classic life cycle. This is the linear sequential SDLC model followed by the following phases Requirement Analysis, Design, Implementation, Testing and Maintenance [1]. The phases are fixed in such a manner the phase once can't be repeated again. As the water fall model is the classical model so it serves as the baseline of all other models. The waterfall model consists of several non overlapping stages as shown in the figure bellow.

It is one way street with no turning back. You cannot move backward once phase "X" is over, you can move to phase "Y". There is no provision of moving backward.

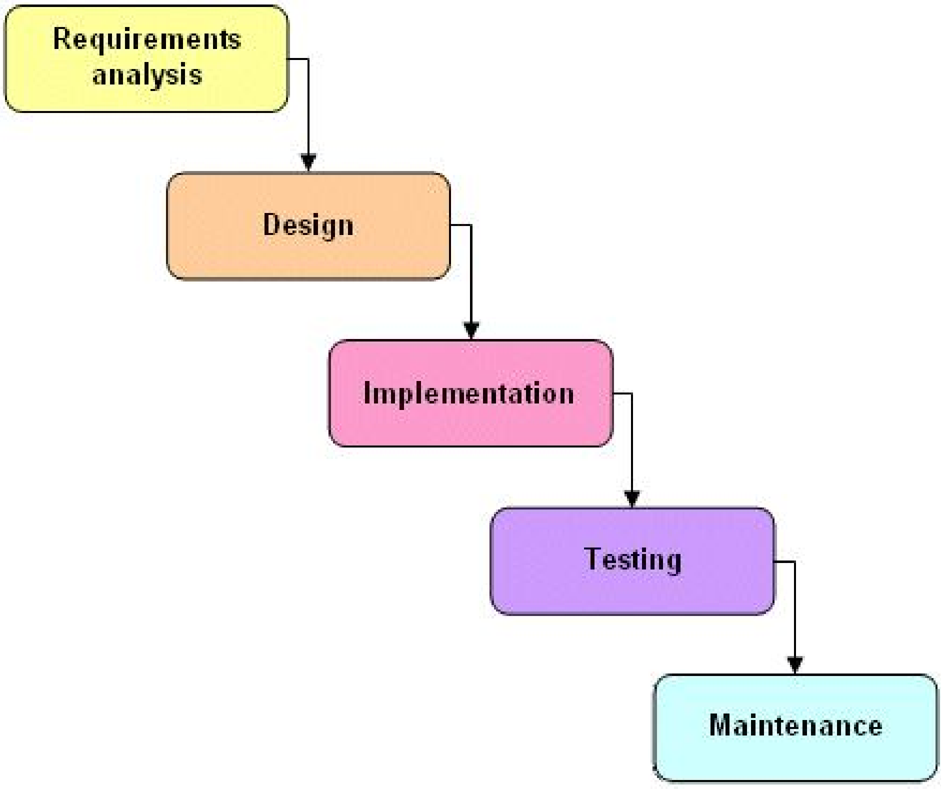


Figure 2 Waterfall model.

Requirement analysis: The software development process starts with the communication between the developer and the customer. According to waterfall model the customer must state all the requirements at the beginning of the project

- **System and Software design:** system and software design includes the complete estimation like cost, time, manpower and scheduling. The complete timeline chart for the project development and tracking.
- **Implementation and unit testing:** It includes the requirements and the project design from developer's side.
- **Integration and system testing:** this phase deals with whether the software meets the specified requirements given by the user.
- **Operations and Maintenance:** After software release this phase is used for enhancement of problems.

Advantages:

I. Easy to understand.
II. Define before design.
III. It is used for project with simple and strict deadlines.

Disadvantages:

I. Idealized.
II. Problems remain uncovered until testing.
III. Requirements are not clear.
IV. Late delivery.
V. Not easy to integrate risk Management.
VI. Difficult to make any changes.

3. The Iterative Waterfall Model

The problems with the classical waterfall model created a demand of new model. The iterative model came into existence to cope with the problems of the original waterfall model the iterative waterfall model is enhanced version of classical waterfall model which could provide faster results require less time and heaving good flexibility. In iterative model project is divided into small parts and this allows developer team to go easily and quickly for their goal and obtain their valuable feedback from users. The project that is divided into small parts here each part is actually a mini waterfall process.

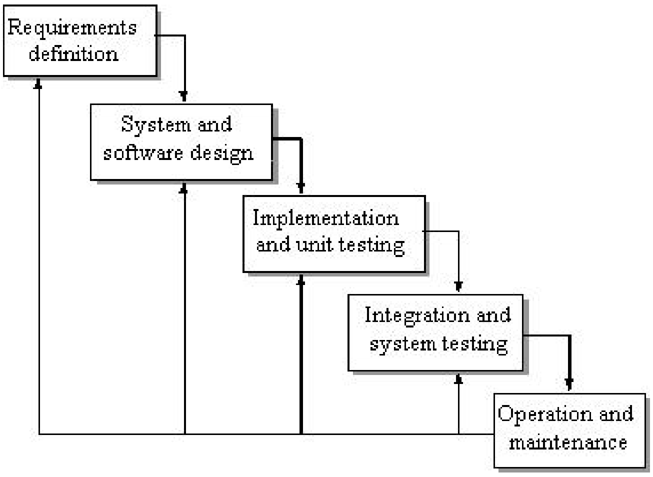

Requirements definition
System and software design
Implementation and unit testing
Integration and system testing
Operation and maintenance

Figure 3 Iterative Waterfall model.

Advantages:

I. Much better model of software process.
II. Client can get Feedback.
III. Used in that type of projects were requirements are not clear.
IV. Document driven process.
V. Works well on week teams.

Disadvantages:

I. Not easy to manage.
II. Not clear mile stones.
III. No stage is finished really.

4. The Prototyping Model

Prototype model is the evolutionary process model. Evolutionary Process models are the iterative type models using this model the developer cam develop increasingly more complete version of software. In prototype model here we have the quick design phase through which the developer has to quickly design the software after requirement gathering as shown in figure.

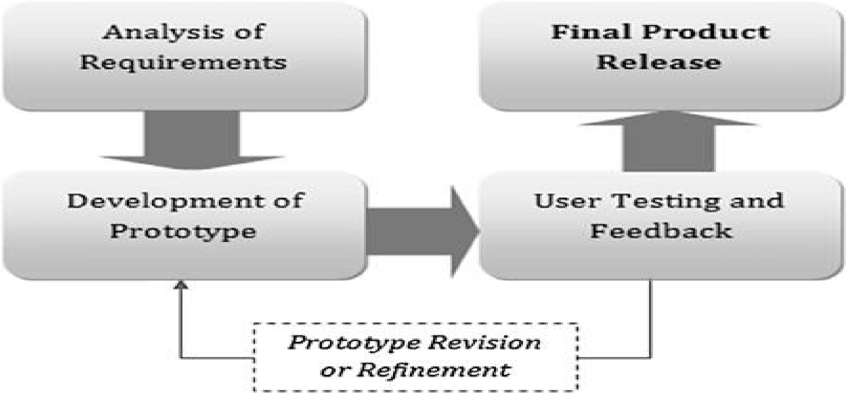


Figure 4 Prototype model

Advantages:

I. Early visibility.
II. Higher outputs.
III. Cost reductive.
IV. Client can actually feel with the system, i.e. feedback.
V. Early design.

Disadvantages:

I. Possibility of unfinished systems.
II. Possibility of inadequate system
III. Lack of flexibility.
IV. Not suitable for large systems.
V. Management is very complicated.

5. The Spiral Model

The spiral model was developed on the end of the 1980s, it was outlined by Barry Boehm, and introduces something that other models did not take into account, which is risk analysis. In essence, the spiral model attempts to bring together key aspects of some other prominent models (namely the waterfall, incremental, and evolutionary prototyping), in an attempt to gather the most appropriate traits from each one, because specific projects might be more or less adaptable to specific models.[3]. It has properties of all above said models. It is also called meta model.

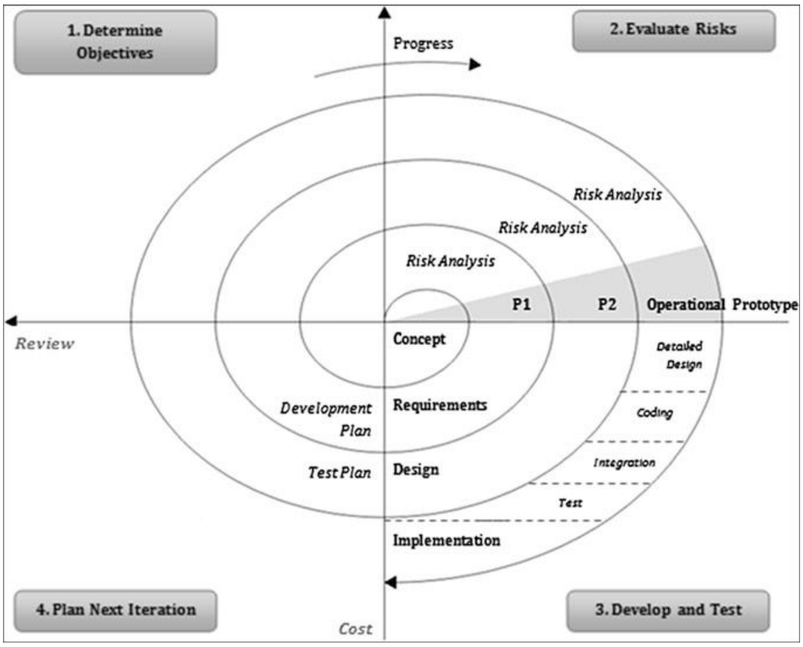


Figure 5 Iterative Waterfall model.

Advantages:

I. New prototype obtained every time.
II. Reuse capabilities.
III. Better productivity.
IV. It has systematic stepwise approach.
V. Elimination of errors in early stage.

Disadvantages:

I. Not proper cost and time estimation.
II. Cost of risk analysis is high on large projects.
III. Complex.

6. The V-Shaped Model

This model can be considered as extension of waterfall model. In waterfall model we move in a linear way while in V shaped model process steps are bent upwards the coding phases to form typical V-Shape [7].
The relationship between each phase of the development process and its associated phase of testing. As testing is one of the important part of Software development the V shaped model emphasis is more on testing. The V shaped Model is almost same as waterfall model both the models are sequential. Requirements have to be very clear before the project starts because it is very expensive to go back to requirements and make changes for the existing projects.

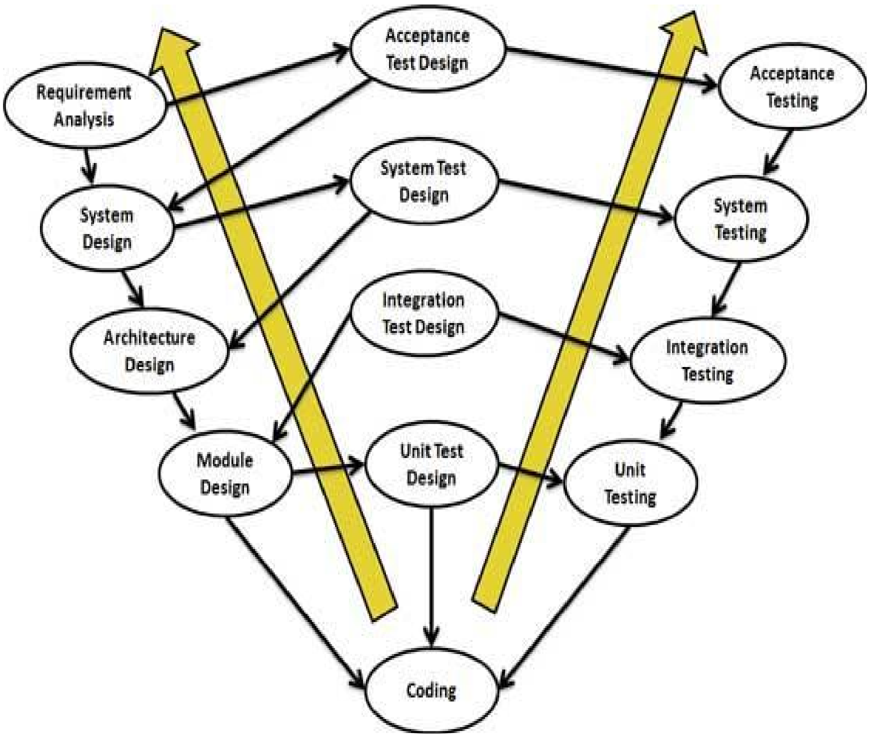


Figure 5 The v- shaped model.

Advantages:

I. High amount of risk analysis.
II. Good for critical projects.
III. Early production.
IV. Easy to manage due to rigidity of model.
V. Easy to understand.

Disadvantages:

I. Not good model for object oriented projects.
II. Not good for long and ongoing projects.
III. Not suitable were requirements have high risk of changing.
IV. Can be costly model.
V. Don’t work well for small projects.

7. Comparative Analysis Of Five Models

MODEL/MERITS	WATERFALL MODEL	ITRATIVE WATERFALL MODEL	PROTOTYPE	V- SHAPED MODEL	SPIRAL MODEL
Success rate	Low	High	Good	High	High
Cost	Low	Low	High	Very high	High
Flexibility	Rigged	Less flexibility	Highly flexible	Little flexible	Flexible
Risk analysis	High	No risk analysis	Low	Low	Low
Cost control	Yes	No	No	Yes	Yes
Reusability	Limited	Yes	Weak	Low	Yes
Risk analysis	Only at beginning	Low	No risk analysis	Low	Yes
Implementation time	Long	Less	Less	Less	Depends on project
Interface	Minimum	Crucial	Curial	Minimum	Crucial
Security	Vital	Limited	Weak	Limited	High
Expertise required	High	High	Medium	Medium	High
Simplicity	Simple	Simple	Simple	Intermediate	Intermediate
User involvement	Only at beginning	At the beginning	High	At the beginning	High
Resource control	Yes	Yes	No	Yes	Yes

8. Conclusion

After completing the analyzing and comparison we conclude that the models are developed from 1970 to 1999. Every model has its own advantages and disadvantages and every model came into existence to cope with the problems of existing model of that time. In now a day’s waterfall and spiral are the most commonly used in the software development process and the other models are used according to the requirements of the software products. The software developers use the models according to the size of the software that is to be developed. All the developers and customers look at the low coat, risk, high quality and small cycle of time, so that the productivity and quality of the software product can optimized.

References

[1]. Barry Boehm, "Spiral Development: Experience, Principles, and Refinements", edited by Wilfred J. Hansen, 2000.
[2]. Roger Pressman, Software Engineering: A Practitioner’s Approach, Sixth Edition, McGraw-Hill Publication.
[3]. K. K. Aggarwal, Yogesh Singh Software Engineering 3rd Edition.
[4]. Karlm, "Software Lifecycle Models', KTH, 2006.
[5]. Vanshika Rastogi ‘Software Development Life Cycle Models-Comparison, Consequences’ ISNN: 0975-9646 International Journal of Computer Science and Information Technologies, Vol. 6 (1) , 2015, 168-172
[6]. Sanjana Taya “Comparative Analysis of Software Development Life” ISSN: 2229-4333 (print), ISSN: 0976-8491(online), vol.2ISSUE 4, Oct-Dec2011

[7]. Nabil Mohammed Ali Munassar And A.Govardhan, “A Comparison between Five Models of Software Engineering” Ijcsi International Journal of Computer Science Issues, Vol7, Issue 5, Sep, 2010.

[8]. Comparative Analysis of Different types of Models in Software Development Life Cycle-Ms. Shikha Maheshwari,Prof. Dinesh Ch. Jain

[9]. JJ Kuhl, "Project Lifecycle Models: How They Differ and When to Use them", 2002 www.businessesolutions.com.

[10]. Rlewallen, "Software Development Life Cycle Models", 2005, http://codebeter.com.

[11]. http://www.tutorialspoint.com/sdlc/sdlc_iterative_model.html

AUTHOR

Manzoor Ahmad Rather did his BCA and MSC-IT from Punjab Technical University Jalandhar Punjab India and Currently he is pursing M.Phil-CS from Maharishi Markandeshwar Institute of Computer Technology & Business Management, M. M. University, Mullana (Ambala). India. His current area of interest is Software Engineering.

Vivek Bhatnagar received his MCA & M. Phil. (Computer Sc. & Technology) degrees and is currently working as Associate Professor at Maharishi Markandeshwar Institute of Computer Technology & Business Management, M. M. University, Mullana (Ambala). His current areas of interest include Software Engineering and Software Reusability. He is an Associate Life Member of Universal Association of Computer and Electronics Engineers (UACEE), Australia, Member of International Association of Engineers (IAENG), Hongkong and Member of the Society of Digital Information and Wireless Communication (SDIWC). He has attended four short term courses/workshops. He has also written a text book titled “Text book of Computers”.