ISSN: 2455 - 9172

EIJO Journal of Engineering, Technology And Innovative Research (EIJO–JETIR)

Einstein International Journal Organization (EIJO)

Available Online at: www.eijo.in

Volume – 1, Issue – 3, July - August -2016, Page No. : 01- 07

Survey on Software Development Life Cycle

Binita Gaurav[1], Sumit Kumar Bola[2], [3]Mrs. Anuradha
[1] M.Tech. Scholar, [2,3]Assistant Professor
Jayoti Vidyapeeth Women's University, Jaipur, India

Abstract

There are various "SOFTWARE DEVELOPMENT LIFE CYCLE" models widely used for developing software. Software Development Life Cycle models are very important for developing the software in systematic manner. Each model has its own characteristics, limitation and working environment. According to the requirements software industry people used different software's. This paper deals with these SDLC models-- namely Waterfall Model, Spiral Model, Iterative Model, V-Shaped Model etc. It allows client and developer to interact freely with each other and implement requirements in a better way .Every SDLC model has its advantages and disadvantages from which we may decide what SDLC model is appropriate for a particular industry as well as a company. In this paper we compare different famous life cycle models.

Keywords: Software, Development, Life Cycle, Comparison, Model, Client, Satisfaction

1. Introduction

Software engineering is a discipline whose aim is the production of quality software that is delivered on time, within budget and that satisfies its requirement. [1]It is very interesting to learn that all the software development industry is based on the specified area. There are various SDLC model widely used for developing software. It allows the client to develop and interact freely with each other in order to understand and implement requirement in a better way with less cost .There are several models for such processes, each describing approach test to a variety of task or activities that place during the process. It aims to be the standard that define the entire task require for developing and maintaining software. [2]

2. Software Development Life Cycle

As we know that Software Development Life Cycle is a process to develop software. These process is divided into some phases such as Requirement Analysis ,Design ,Coding , Testing and Maintenance .All these activities are carried out in different ways as per the client's need. Each way is known as a SDLC Model. [3][4] **Requirement Analysis**: This phase is the main focus of the project manager and stack holder. Meeting with Managers, Stack holders and users are held in order to determine the requirements like.

Who is going to use the system? And how will they use the system?

After gathering requirements these are analyzed for their validity and possibility of the system. These are also to be developed and studied finally for a requirement specification document which is created with series of the purpose of guidelines for the next phase of the model .It is more crucial phase in SDLC [5][6].

Design: System design in specifying hardware and software requirements also helps in defining over all architecture system. The system design specification serves as input for the next phase of the model.

Coding: On receiving system design documents the work is divided in modules unit and then only actual coding is started .In this phase the code is produced therefore it becomes the main focus for the developer. This is the longest phase of the software developer life cycle.

Testing: Testing is the most important and powerful phase. Effective Testing will contribute to the delivery of high quality users with a low maintenance cost and reliable results. [6]

Maintenance: This phase is started after the delivery of the product. If any error is occurred or modification is needed in it then implementation is required in this phase.

3. SDLC Model: There are five kinds of SDLC MODEL

- Waterfall Model
- Spiral Model
- Iterative Model
- V-Shaped Model

Waterfall Model: The Waterfall Model is the classical model of software engineering. This model is one of the oldest models and is widely used in government projects as well as in many major companies. All works are divided into five different phases .All phases are cascaded to each other so that the second phase is started as and when defined set of goals are achieved. When the first phase is signed off thus it is named as "WATERFALL MODEL". Phases of waterfall model have been already discussed above in this paper .

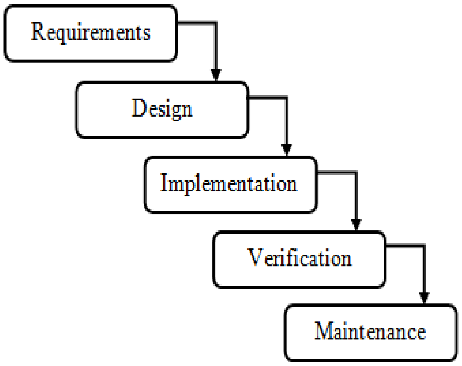


Figure 1: Phases of Waterfall Model

Now we may discussed the basic principles of Waterfall Model:

* Project is divided into sequential phase .Sometimes it will overlap and splash back which are acceptable between the phases.

*Emphasis is on planning, time schedule, target dates, budget and implementation of the entire system at one time.

Advantage:

1. Simple and easy to use.
2. Easy to arrange tasks and define stages clearly.
3. Requirements should be clear before going to the next phase.

4. Each phase of development proceeds in linear order without any overlapping.

5. Works for project where requirements are understood.

Disadvantage:

1. Users may judge quality only at the end.

2. It does not allow any change required by the client.

3. Heavy amount of risk.

4. Users do not get the full products before delivery.

5. It follows the “BIG- BANG “approach. The entire software is delivered in one slot at the end.

Spiral Model: This model was not the first model [7] but it was the first model to explain why the iteration matter .Each phase starts with design goal and ends with the client (who may be internal) reviewing the progress . Analysis and engineering efforts are applied at each phase of the project. It moves clock wise in traversal each in a deliverable. [8] This change is from traversal to traversal .The development team starts with a small set of requirement and goes through each development phase (except installation and maintenance) for those set of requirement. [9]

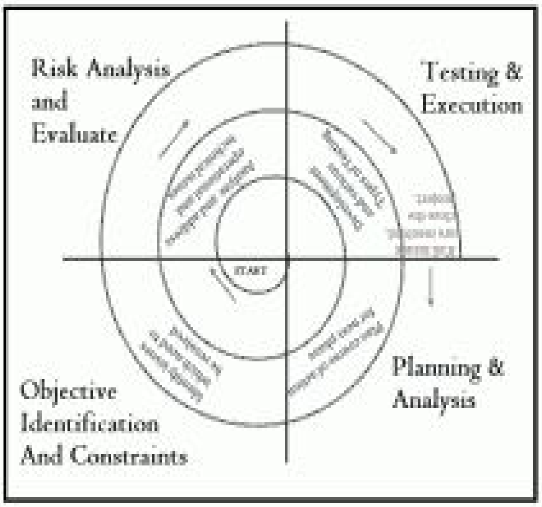

Risk Analysis and Evaluate
Testing & Execution
START
Planning & Analysis
Objective Identification And Constraints

Figure 2: Spiral Model

For Example: The first traversals may result in a requirement specification. The second will result in a prototype and the next one will also result in another prototype or sample of a product until the last traversal leads to a product which is suitable to be sold .Consequently the related activities and their documentation will also mature towards the outer traversals.

For example: A formal design and testing session would be placed in to the last traversal [10].

Advantage

1. High amount of risk analysis and avoidance of risk enhanced.

2. Good mission for large and critical projects.

3. Strong approval and documentation control.

4. Additional functionality can be added at a later date.

Disadvantage

1. It is a costly model to use.

2. Risk analysis requires highly specification of specific enterprise.

3. Project success is highly depended on the risk analysis phase.

4. Doesn’t work for small projects.

Iterative Model: An Iterative Model does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which can then be reviewed in order to identify further requirements. This process is repeated producing a new version of the software for each cycle of the model. [11]

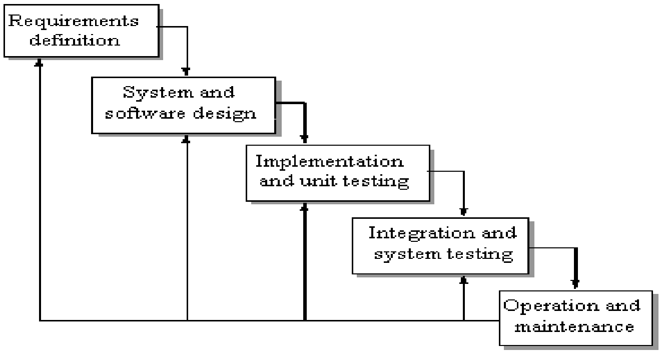

Requirements definition
System and software design
Implementation and unit testing
Integration and system testing
Operation and maintenance

Figure 3: Phases of iterative model

Basic Principle

The problem 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 after greater flexibility. Iterative Model , the project is divided into small parts .This allow 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 virtual information for the design of the next phase. [2]

Advantage

1. In Iterative Model we are building and improving the product step by step .Hence we can track the defects at early stage this avoids the down word flow of the defects.

2. In Iterative Model less time is spent on documenting and more time is given for designing.

Disadvantage

1. Requirements of the computer system are clearly defined and understood

2. When the project is big.

3. Major requirement must be defined ; however , some details can in value with time [11]

V-Shaped Model: V-Model means verification and validation model. Just like the waterfall model, the V-Shaped life cycle is a sequential path of execution of processes. Each phase must be completed before the next phase begins. Testing is emphasized in this model more than the waterfall model. The testing procedures are developed early in the life cycle before any coding is done, during each of the phases preceding implementation. Requirements begin the life cycle model just like the waterfall model. Before development is started, a system test plan is created. The test plan focuses on meeting the functionality specified in requirements gathering. The high-level design phase focuses on system architecture and design. An integration test plan is created in this phase in order to test the pieces of the software systems ability to work together. However, the low-level design phase lies where the actual software components are designed, and unit tests are created in this phase as well.

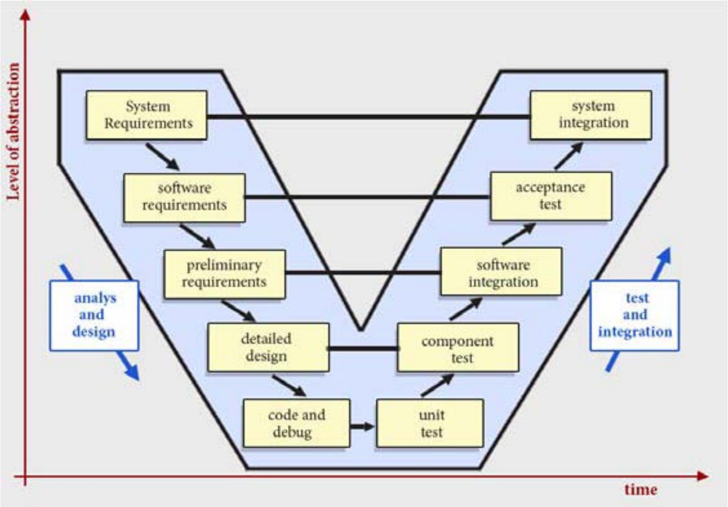


Figure 4: Phases of V-Shaped Model

Basic Principle

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 to occurring at the end of most phases before beginning the next phase. [2]

Advantage

1. Simple and easy to use.

2. Each phase has specific deliverables.

3. Higher chance of success over the waterfall model due to the early development of test plans during the life cycle.

4. Works well for small projects where requirements are easily understood.

Dis Advantage

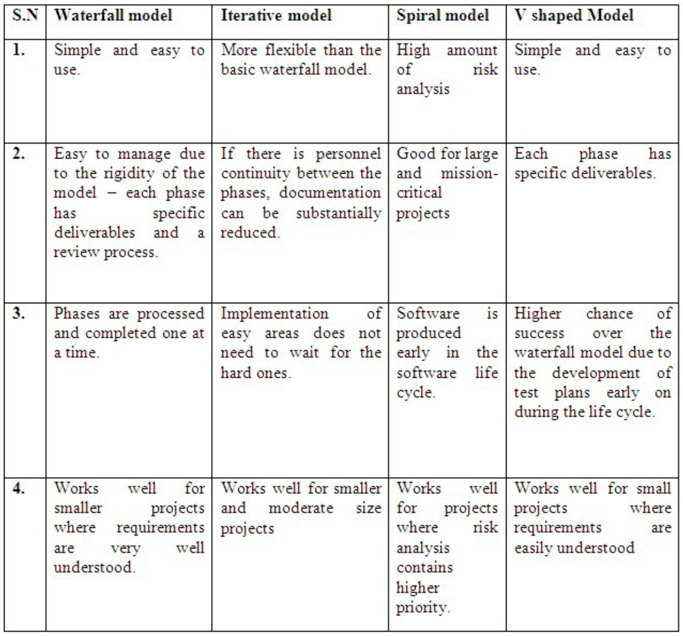
1. Very rigid like the waterfall model.

2. Little flexibility and adjusting scope is difficult and expensive.

3. Software is developed during the implementation phase, so no early prototypes of the software are produced.

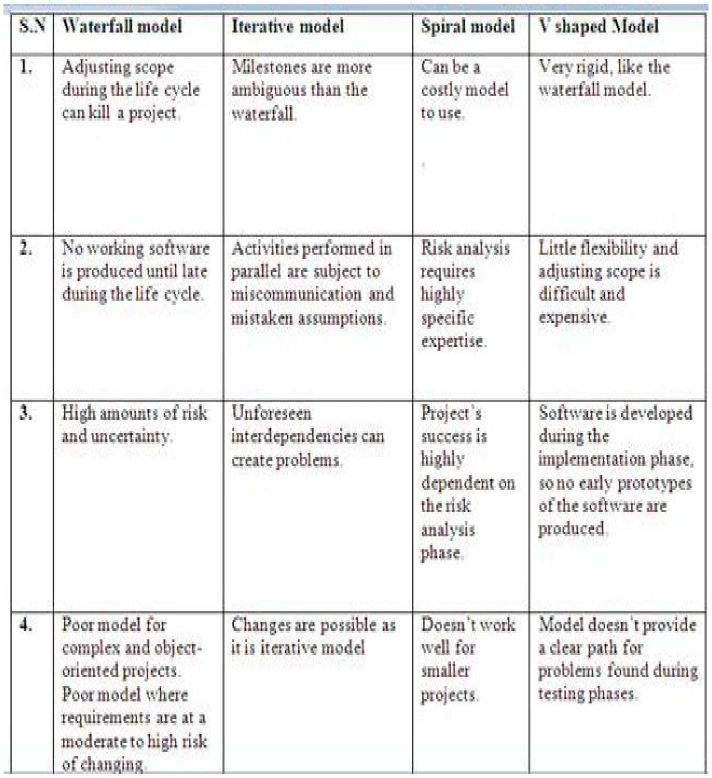
4. This Model does not provide a clear path for problems found during testing phases [7].

TABLE .1 Comparison of Advantage



S.N	Waterfall model	Iterative model	Spiral model	V shaped Model
1.	Simple and easy to use.	More flexible than the basic waterfall model.	High amount of risk analysis	Simple and easy to use.
2.	Easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.	If there is personnel continuity between the phases, documentation can be substantially reduced.	Good for large and mission-critical projects	Each phase has specific deliverables.
3.	Phases are processed and completed one at a time.	Implementation of easy areas does not need to wait for the hard ones.	Software is produced early in the software life cycle.	Higher chance of success over the waterfall model due to the development of test plans early on during the life cycle.
4.	Works well for smaller projects where requirements are very well understood.	Works well for smaller and moderate size projects	Works well for projects where risk analysis contains higher priority.	Works well for small projects where requirements are easily understood

TABLE 2



S.N	Waterfall model	Iterative model	Spiral model	V shaped Model
1.	Adjusting scope during the life cycle can kill a project.	Milestones are more ambiguous than the waterfall.	Can be a costly model to use.	Very rigid, like the waterfall model.
2.	No working software is produced until late during the life cycle.	Activities performed in parallel are subject to miscommunication and mistaken assumptions.	Risk analysis requires highly specific expertise.	Little flexibility and adjusting scope is difficult and expensive.
3.	High amounts of risk and uncertainty.	Unforeseen interdependencies can create problems.	Project's success is highly dependent on the risk analysis phase.	Software is developed during the implementation phase, so no early prototypes of the software are produced.
4.	Poor model for complex and object-oriented projects. Poor model where requirements are at a moderate to high risk of changing.	Changes are possible as it is iterative model	Doesn't work well for smaller projects.	Model doesn't provide a clear path for problems found during testing phases.

4. Conclusion

There are various models of software development life cycle. The aim of software engineering is to develop software high quality within budget and schedule each has its own advantage or disadvantage depending upon which we have to decide which model we should choose. In this paper we have compared the different software development life cycle models on the basis of some advantages or disadvantages. On the basis of this advantage for a particular software project one can decided which SDLC should be chosen first that particular project. We can see the V – Model is used because V-Model involves verification correctness of software. Every stage of development life cycle. It involves continuously testing the system during all stages of the development process rather than just limiting testing to the last stages V-Model enables testing from initial stage.

5. Future Work

This paper focused on the existing models. There are various shortcomings in the existing models; in future we can have models that can overcome the drawbacks of the existing models.

6. References

[1]. K. K. Agarwal, Yogesh Singh Software Engineering 3rd Edition. Comparative analysis of different types of model in Software Development Life Cycle.
[2]. Software Development Life Cycle (SDLC) – the five common principles.htm
[3]. Software Methodologies Advantages & disadvantages of various SDLC models.mht
[4]. www.shazsoftware.com/software-development-life-cycle.html
[5]. www.waterfall-model.com/sdlc/
[6]. weblog.erenkrantz.com/~ jerenk/phase-2/Boe88.pdf
[7]. www.ccs.neu.edu/home/matthias/670s05/lectures/2.html
[8]. Vishwas Massey, K.J Satao, "Comparing Various SDLC Models And The New Proposed Model On The Basis Of Available Methodology".
[9]. Software engineering models consequences and. Alternatives Nitin Mishra , Shantanu Chowdhary, Arunendra Singh , Anil Sharma.
[10].http://istbexamcertification.com/what-is-iterative-model-advantages-disadvantages-and-when-to-use -it/
[11].http://istbexamcertification.com/what-is-V-Shaped model-advantages-disadvantages-and-when-to-use -it/