



ISSN: 0976-3031

Available Online at <http://www.recentscientific.com>

CODEN: IJRSFP (USA)

International Journal of Recent Scientific Research
Vol. 10, Issue, 07(J), pp. 33957-33961, July, 2019

**International Journal of
Recent Scientific
Research**

DOI: 10.24327/IJRSR

Research Article

A BRIEF SURVEY ON SOFTWARE DEVELOPMENT LIFE CYCLE MODELS

Shashi Prabha Anan¹., Esha Purwar²., Seema³ and R. B. Gautam⁴

^{1,2}Department of Computer Science & Engineering, MCAET, Ambedkar Nagar, India

³Department of Civil Engineering, MCAET, Ambedkar Nagar, India

⁴Department of Agriculture & Engineering, MCAET, Ambedkar Nagar, India

DOI: <http://dx.doi.org/10.24327/ijrsr.2019.1007.3788>

ARTICLE INFO

Article History:

Received 4th April, 2019

Received in revised form 25th May, 2019

Accepted 23rd June, 2019

Published online 28th July, 2019

Key Words:

Software Performance, Verification, Validation, Risk analysis, Customer Evolution, Prototype model etc.

ABSTRACT

Software Development Life Cycle Models (SDLC) are systematic and well defined approach for problem solving; it is a series of identifiable stages that software goes during its lifetime. It practiced for developing high quality and reliable software system. In the early years, software was cheaper than hardware. In digital era, hardware is cheaper and software is expensive. So the costs of hardware and software have been reversed due to increased demand of well engineered software products so we have to develop software products due to increasing demand of software in market so for developing a software product we have to follow the models that deliver a quality software product. There are many SDLC models available, but in this paper we make our attention in four SDLC models namely; Waterfall model, V shaped model, Spiral model, and Prototype model, there uses, advantages and disadvantages.

Copyright © Shashi Prabha Anan et al, 2019, this is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution and reproduction in any medium, provided the original work is properly cited.

INTRODUCTION

Every work is done with the help of a computer in real life, so we need to develop a software system that meets all user needs and is easy to use. Thus SDLC systematically and disciplined develops the project. It defines entry and exit criteria for each phase, a phase can only begin if the entry and exit criteria are met so that it becomes very difficult without using SDLC models.

A software design strategy relates to the structure used to plan, manage and control the method of creating a software system [1]. Formally, the methodology of software development is known as the SDLC abbreviate as the Software Development Life Cycle and is used in many areas, for example, engineering of software, system engineering, computation and other industrial applications.

For example, software engineering, system engineering, informatics engineering, mechanical engineering and applied engineering [2].

Software was inexpensive in the early days and equipment was expensive. But now day software is costly and hardware is cheap. So the software and equipment expenses have been inverted owing to growing demand for well-designed goods.

Motivation of SDLC Models

- Encourage computer developers to use multiple SDLC designs for well-maintained initiatives.
- Encourage consciousness of software performance.
- To research the facts that influence software performance.

Phases of Software Development Life Cycle Models

Series of identifiable stages that software goes during its life cycle are:

1. Feasibility study
2. Requirement analysis
3. Design
4. Coding
5. Testing
6. Maintenance

Feasibility study: It works out on overall understanding of problem, formulates different solution and strategies, performs a cost benefit analysis, may also find none of solution is feasible due to high cost, resource constrains, technical reason.

Requirement analysis: Aim of this phase to understand the exact requirement of customer then document them properly.

*Corresponding author: **Shashi Prabha Anan**

Department of Computer Science & Engineering, MCAET, Ambedkar Nagar, India

Design: Design phase requirement specification is transformed into a suitable form.

Coding: During this phase each module of the design is coded.

Testing: During this phase different modules are integrated in a planned manner then testing is done.

Maintenance: Correct error which was not observed during the product development phase.

LITERATURE REVIEW

This section contains various models their advantages, disadvantages and when to use them:

Waterfall Model

In 1991, David Whitgift [3] points out that software was first published and then debugged in the previous days. After finishing one stage, we're going to another stage; there was no official method to layout assessment.

Kai Petersen et. in 2009. AI.[4] Conduct a case research to provide more comprehensive descriptions of the problems and identify four fresh problems, namely-

1. Which part of the necessity is enforced by who is a question of Confusion?
2. Great maintenance effort is needed
3. Focus on specialized skills and people with less trust
4. Due to the communication obstacle, it is very tedious to find the fault.

In 1970, Winston W. Royce [5] points out The Waterfall system describes several successive stages that must be finished one after the other and moved to the next stage only when its previous stage is complete. For this reason, the model of the Waterfall is recursive in that each phase can be repeated endlessly until it is perfected.

Fowler [6] states that waterfall model splits up projects depending on operations: request assessment, layout, coding, and testing.

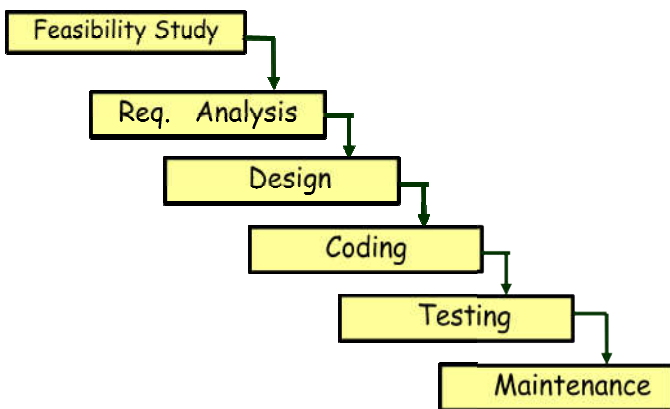


Figure 1

In fig (1) we can say that there are six phase phases between feasibility study and testing is called development phase, among all life cycle maintenance phase consumes maximum effort, among development phase testing phase consumes maximum effort.

Advantages

1. It's simple to comprehend and use
2. Provides a reference to inexperienced personnel.
3. Team understands milestones well.
4. Provides stabilization of the requirements.

Disadvantages

1. All specifications must be understood in advance.
2. Can offer a mistaken feeling of advancement.
3. In the end, integration is a large bang.
4. Little chance for customers to test the system.

When to use Waterfall Model

1. Requirements are well known and stable.
2. Technology is understood.

V Model

V-model means a verification and validation scheme. Like the waterfall model, the V-Shaped lifecycle is a continuous process implementation route. Before the next stage starts, each stage must be completed. Product testing is planned in combination with the corresponding layout phase.[7], its emphasis on verifying the validity of V&V activities spread over the entire lifecycle.

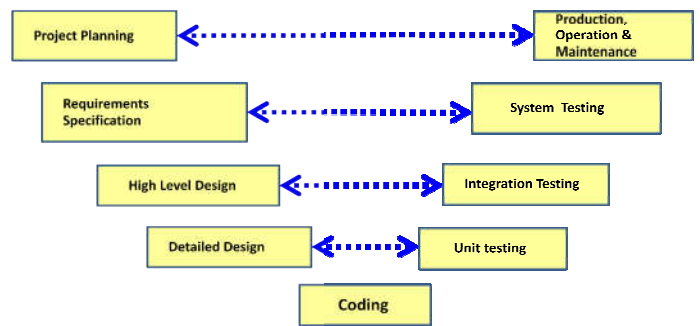


Figure 2

In fig (2) we can see that in every phase of development testing activities are planned in parallel and development.

Advantages

1. This system is very simple and user-friendly.
2. Testing operations such as scheduling, sample designing, is done well before coding that saves plenty of time. Hence a greater chance of achievement over the waterfall model
3. Defects discovered at an early point that can be easily tracked and solved.
4. This design is kept away from the defects ' downward stream.
5. This design is used in tiny projects where demands are easily understood.
6. Every deliverable is created testable.
7. Facile to use.
8. Starting from the early phases of growth.

Disadvantages

1. Software is created during the application stage, so no early software prototypes are generated.

2. If any modifications occur in midway, then the test documents along with requirement documents have to be updated.
3. Very stiff and the least versatile
4. Does not promote phase overlap.
5. Does not provide assistance for efficient risk management.
6. Does not manage stages or iteration.

When To Use V Model

1. Natural choice for system requiring high reliability.
2. All requirements are known upfront.
3. Solution and technology are known.

Prototype Model

“... The assumption that one can specify a satisfactory system in advance, get bids for its construction, have it built, and install it. ...this assumption is fundamentally wrong and many software acquisition problems spring from this...” Frederick Brooks

The existence of the prototype that the customer examines avoids many misunderstandings that happen when each party believes that the other knows what they're saying. The end item is more probable to fulfill the look, feel and execution requirement of the user. Incremental model is at the core of the design method of cyclic software. It begins with an original preparation and finishes with the cyclic relationships between them being deployed. Easier in a larger iteration to test and debug. Easier to handle danger due to the identification and handling of dangerous parts during its iteration.. Spiral model is useful for big and mission-critical initiatives where elevated threat assessment such as satellite launch is required [8].

The basic idea here is that instead of freezing the requirements before a design or coding can proceed, a throwaway prototype is built to understand the requirements. This prototype is created on the basis of the presently recognized demands. By using this prototype, the client can get an "actual feel" of the system as the interactions with the prototype can enable the client to better understand the requirements of the desired system. Prototyping is an appealing concept for complex and big applications for which there is no manual method or current scheme to assist determine the demands. Usually, the prototype is not full systems and many of the information are not integrated into the prototype. The objective is to provide overall functionality to a system[9].

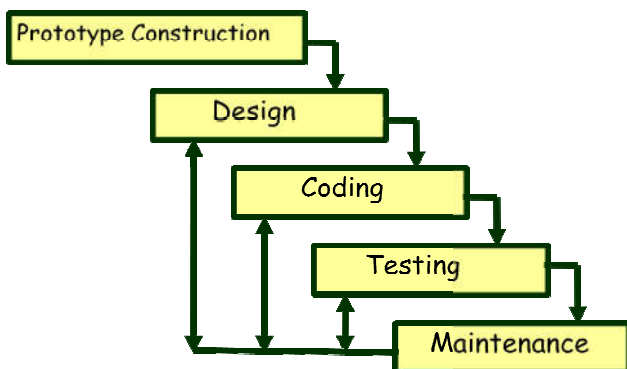


Figure 3

Advantages

1. This system is very simple and user-friendly.
2. Testing operations such as scheduling, sample designing, is done well before coding. That saves a ton of moment. Hence a greater possibility of achievement over the template of the cascade.
3. Defects discovered at an early point that can be readily tracked and solved.
4. This design is kept away from the defects ' downward stream.
5. This design is used in tiny initiatives where requirements are readily understood.
6. The user's demands are better suited.
7. The structure is of the highest standard.
8. It is simpler to keep the resulting software.
9. Overall, the price of growth is lower.

Disadvantage

1. Susceptible to over engineering. rigid and least flexible
2. Software is developed during the implementation phase, so no early prototypes of the software are produced.
3. If any changes happen in midway, then the test documents along with requirement documents has to be update
4. The V-shaped model should be used for small to medium sized projects where requirements are clearly defined and fixed.
5. The V-Shaped model should be chosen when ample technical resources are available with needed technical expertise.
6. For some project its expensive

When to use Prototype Model

1. Prototype model should be used when the required scheme requires a bunch of end user communication.
2. Typically, internet applications, internet applications have a very elevated level of communication with end customers, are best suited for prototype model Building a scheme that enables ease of use and requires minimal preparation for the end customer may take a while.
3. Prototyping guarantees that end consumers are continuously working with the scheme and providing input that is integrated into the prototype to result in a usable scheme. They are outstanding for the design of good human computer interface structures

Spiral Model

The launch of the item could be a prototype or a paper model throughout the original iterations. But more and more complete versions of the software applications are being generated during sophisticated iterations. As illustrated in fig. 2.1, This design is split into a collection of frameworks. These operations represent every section of the spiral route As the design method starts, the software crew conducts duties suggested in a diagonal manner by the loop around the Spiral. This model is a realistic method to large-scale project growth. As the software development method progresses, the client & developer reacts and understands the hazards better at each stage of the Spiral system.

Similar to the incremental model, the spiral model [11] places more emphasis on threat assessment. There are four stages in the spiral model planning, risk analysis, engineering and evaluation in iterations (called spirals in this model a software project constantly goes through these stages. The baseline spiral which starts in the planning phase collects the demand and evaluates the risk each spiral that follows builds on the baseline Spiral. During the planning stage, requirements are collected. A method is conducted in the risk assessment stage to define risk and alternative solutions at the end of the threat assessment stage, a prototype is produced in the engineering stage, and software is generated along with end-of-phase inspection. The evaluation phase enables the client to assess the project's performance to information before proceeding to the next spiral. The angular element in the spiral model reflects advancement, and the spiral radius reflects price each loop of the spiral represents a phase of s/w process:

- The innermost loop might be concerned with system feasibility
- The next loop with system requirement definition.
- The next one with system design and so on.

There are no fixed phases in this model; the phases shown in figure are example the team must decide how to structure the project into phases each loop of the spiral model is split into four sectors (quadrants) [12]:

First quadrant (Objective Setting)

- During the first quadrant, we need to identify the objectives of the phase.
- Examine the risks associated with these objectives

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

Third quadrant (Development & validation)

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

Fourth quadrant (review and planning)

- Review the results achieved so far with the customer and plan the next iteration around

The spiral.

- With each iteration around the spiral, progressively a more complete version of the

Software gets built.

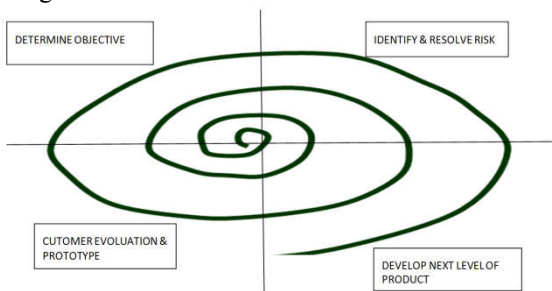


Figure 4

Advantages

1. Additional functionality or modifications can be made at a subsequent point
2. Cost assessment becomes simple as prototype construction is performed in tiny pieces
3. Continuous or recurring implementation aids in risk management
4. Development is rapid and characteristics are added systematically
5. Customer feedback is always available.

Disadvantages

1. Risk of failure to meet timetable or budget
2. It operates best for big initiatives just as it requires knowledge in risk evaluation
3. Strictly must be pursued for its clean operation loop design protocol.
4. Documentation is more important because it has intermediate stages
5. It's not recommended for a bigger venture, it could cost them

When to use Spiral Model

1. When project is large
2. When releases are required to be frequent
3. When creation of a prototype is applicable
4. When risk and costs evaluation is important
5. For medium to high-risk projects
6. When requirements are unclear and complex
7. When changes may require at any time
8. When long-term project commitment is not feasible owing to modifications in financial priorities

Comparison

The comparison between different models is shown by their advantages and disadvantages:

Table 1 comparison of waterfall, v model & spiral model [13]

Waterfall Model	V Model	Spiral Model	Prototype Model
It is simple and easy to use.	It is simple and easy to use.	There is too much amount of risk analysis.	It is complex.
Easy to maintain due to its rigidity each phase has deliverable process.	Each phase has specific deliverable.	Good for large projects.	Good for big and complicated projects.
Each phase are processed and completed one at a time.	High chances of success over waterfall due to the early testing.	Software is produced early.	Time consuming.
Works well for small project where requirement and technology are understood.	Works well for project where security is important.	Works well for small project where requirement are understood.	Works for designing good human computer interface systems

Table 1 comparison between waterfall and prototype model [13]

Table 2 Comparison between waterfall and spiral model

Waterfall model	Prototype model
User can only preview the system only after the final version of the software is developed because there is no feedback path. This is easy to understand. Complete program analysis It uses sequential approach. Waterfall Model is implemented in the projects where conventional product/software has to be delivered to the client. In such cases user is sure and clear about his requirements. He states them clearly to the development team and chances of ambiguity is bare minimum	User have a preview of the system from the "quick design" and the prototype developed early at the of the process This is not easy to understand. Incomplete program analysis. It uses evolutionary approach.
Works well for small projects. Woks well for project where requirement are well defined.	Prototype model is usually used in online projects where client is not sure about his requirements, his expectations and preferences need to be taken care of. Works well for both small and large projects. Can also works for project where requirements are not clear.

CONCLUSION

In this document we address waterfall model, v model, spiral model, prototype model of their advantages disadvantages, when to use them and also their comparative charts Waterfall model offers a basis for other model and it performs well where technique is known, v model works well for security function software, prototype model works well for big initiatives and spiral model works well for threat assessment. We can use any model according to our needs and create high-quality software.

Future Scope

In the future we can change designs according to their requirements and we can take advantage of all designs and create a fresh template with different characteristics of Waterfall model, V model, Spiral model and Prototype model.

References

1. Ian Sommerville, Software Engineering, Addison Wesley, 9th ed., 2010
2. Richard H. Thayer, and Barry W. Boehm, "software engineering project management", Computer Society Press of the IEEE, pp.130, 1986
3. Whitgift, David, "Methods and Tools for Software Configuration Management", J. Wiley, 1991.
4. Petersen K., Wohlin C., Baca D. , "The Waterfall Model in Large-Scale Development", Lecture Notes in Business Information Processing, vol 32. Springer, Berlin, Heidelberg, 2009.
5. Royce, W., "Managing the Development of Large Software Systems", Proceedings of IEEE WESCON 26, pp.1-9, 1970
6. Fowler, M. (2004), UML Distilled a Brief Guide to the Standard Object Modelling Language, Boston: Pearson Education, Inc.
7. "istqbexamcertification.com/what-is-v-model-advantages-disadvantages ", by S Kumar - Jan 13, 2012.
8. 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
9. istqbexamcertification.com/what-is-prototype-model-advantages-disadvantages", by S Kumar -Jan 12, 2012.
10. Roger S.Pressman, "Software engineering: A practitioner approach", ISBN 0-07-365578-3,5th ed., TMH, 2001
11. Boehm B, "A Spiral Model of Software Development and Enhancement", ACM SIGSOFT Software Engineering Notes", "ACM", 11(4):14-24, August 1986.
12. Ian Sommerville, "Software Engineering", Addison Wesley, 9th ed., 2010
13. Study & Comparison of Software Development Lifecycl ModelsGourav Khurana, Sachin Gupta

How to cite this article:

Shashi Prabha Anan *et al.* 2019, A Brief Survey on Software Development Life Cycle Models. *Int J Recent Sci Res.* 10(07), pp. 33957-33961. DOI: <http://dx.doi.org/10.24327/ijrsr.2019.1007.3788>
