

# A Review on Software Models

S Reshma<sup>1</sup>, G. Shaheen Firdous<sup>2</sup>, Dr.M.Jahir Pasha<sup>3</sup>

<sup>1</sup>Assistant Professor in Computer Science and Engineering department at  
Dr. K V.Subba Reddy College of Engineering for Women, Kurnool

<sup>2</sup>Assistant professor in Computer Science and Engineering department at  
Dr. K V.Subba Reddy College of Engineering for Women, Kurnool

<sup>3</sup>Associate Professor & HOD-Computer Science and Engineering department at  
Dr. K V.Subba Reddy College of Engineering for Women, Kurnool

## Article Info

Volume 83

Page Number: 2033 - 2037

Publication Issue:

March - April 2020

## Abstract

The idea of managing the working domain in the overall world is changing very often in a fast manner. The world is forwarding to more and more sophisticated and techno nature one from traditional way of working domain. To appeal to this new transformation, the companies were proceeding with the resource of advanced technology and considerable employing computer systems. In addition, acquiring the new technology, shifting to Software development has become principal means of undertaking business in the current trend. The increasing need of software development has steps to the growth of IT industry throughout the world. The main cause of reversing into software development is to create value and growth to the requirement of the users. Basically, software development should serve the user to fulfill their requirements in marketing domain and to develop a better software application to routinize this requirement. The increasing user requirements are continuously witnessed in the recent days and thus the necessity of software trend development is also increasing ever. We are witnessing dazzling growth of IT industry as of the higher requirement software development which is being provided for by many number of IT consultancy companies. The discipline of software development covers the number of forms involving such a complicated operation, serving huge cases of industries and it covers all kinds of human activities we could think of. Many offices are attempting to make their operation paperless and software development is playing a leading role to achieve this target. Therefore the growth of much company has become much dependent in the present world, with more and more development efforts.

## Article History

Article Received: 24 July 2019

Revised: 12 September 2019

Accepted: 15 February 2020

Publication: 18 March 2020

Keywords: Software, IT industry.

## 1 : INTRODUCTION

Software Development Life Cycle (SDLC)

SDLC is defined as a well-defined and a structured chain of orders in software engineering so as to motivate an intended software product [4].

- Communication

The initial step that the user ignites the request found available for the desired software product. Secondly, with the advice of service provider and the terms present are eliminated. Then the requests are selected to provide service passing organization of writing.

- Requirement Gathering

The gathering is a step from which the software development team functions to start the project at once. A clear discussion is made with various storage from the problem sequence and attempts to conduct out as much as possible information from their requirements. These requirements are inspected and separated into series of requirements from the user, system and functional requirements. Then they are collected through a various number of practices as follows,

1. Learning the existing or outdated system and software levels,
2. Managing the interviews of the users and developers,

3. Mentioning or referring to the database,
4. Grabbing answers from the list of questionnaires.
5. Study of feasibility.

The software team arises with a set of pseudo plan of the software process. In this stage, the team checks the quality of software is possible to be produced to satisfy the expectation given by the user and analyze is there any useless software is being used by the user. It is found that if a project is practically, financially and technologically reasonable or feasible for process any function in an organization. In addition, there are many number of algorithms is present, which help to reach the goal to attain feasibility from the software developers.

- System Analysis

In this step, the developers design a roadmap to achieve their plan and attempt to bring up the perfect software model that is suitable to start the project. This process includes the Knowledge from software analysis like drawbacks, problems in the learning system or alterations to be accomplished in the existing systems as soon as possible. Then the identification is analyzed and the impact of project in an organization is addressed in a separate way. The team involved in the project searches the scope and aim of the project and schedule plan and resources according to the project.

- Software Design

The knowledge requirement is retrieved once and details are understood, then the design of software product is initiated. The inputs from users and gathered information in requirement collection phase are taken as the inputs for this step. The output provided by this step arises in the form of designs such as logical and physical design. The Software engineers try to produce meta-data information and the objects such as dictionaries, logical diagrams and flow-charts of

data and also the pseudo codes of the system design.

- Coding

Coding can be called as programming or simply organizing steps in an operating system. The code implementation begins with the means of creating the program code in a suitable or appropriate programming language and growing the errorless executable programs in an efficient manner.

- Testing

The estimation of this phase deals with the 50% test in the software development criteria. The errors may decay the software from its critical level to self-removal. The process software testing is processed while coding is developed by programmer and full system test is taken by some experts in testing at different levels of program code like testing of the module, program, product, in-house and the product at user's end. Early invention and detection of errors and problems, in which the rescue is the key to reliable software solution.

- Integration

Integration is a necessary step to be performed, in which the software is needed to be integrated with the corresponding libraries, databases and other the programs found. The integration stage in SDLC is involved in the inclusion of software with the outer world operations.

- Implementation

The installation of software on the user machines are performed in this stage of implementation. In this stage, the software requires post installation or configurations at the user end. The fully designed software is verified and inspected for the issues related to characters such as portability, system adaptability, also integration methods that are solved during this process.

- Program operation

This process confirms the operation of software, in both the terms like improved efficiency and reduced errors. That software is handled timely by the updation of code related to the variations or changes present in the user end system or technology. The operation phase is noticed to face many challenges such as hidden bugs and many real-world unrecoverable problems.

- Disposition

After some time, the software may reject the performance in the front phase itself. It may forward completely confused or it may need an intense up-gradation [5]. Thus a impressions need to remove the major portion of the raised system. This also includes the process called archiving of data and wanted software components, shutting down the system, planning misplaced position activity present and terminate of system at concurrent end-of-system time.

## 2: SOFTWARE MODELS

It allows the developer to choose a strategy that is suitable to develop the software. This paradigm consists of individual methods, tools, and procedures that are expressed legibly and it defines a SDLC process. Some of the of software paradigms and related process models are given below

### 2.1 Waterfall Model

This is the simplest and précised model present in the SD paradigm or simply SDLC. It gives that all the phases of SDLC functions one after the other in a linear fashion. Likewise, when the first phase is completed, then the second phase is ready to start and other processes follow.

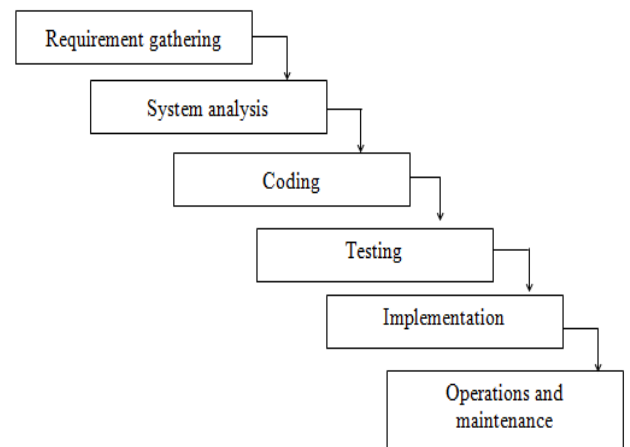


Figure 1 Waterfall Model

The model, in which all the process has been carried away perfectly as planned in the early stages without a need to think about past problems, which may come in the future phase. This model fails to work smoothly when there are some issues left during the previous stage. Thus, this model fits good for the developers who already have designed and built similar software in the past and those aware of its entire domains.

### 2.2 Iterative Model

The model brings the SD process in the way or mode of iterations. It involves projecting the development process in a cyclic manner by repeating each and every step after each cycle of SDLC method.

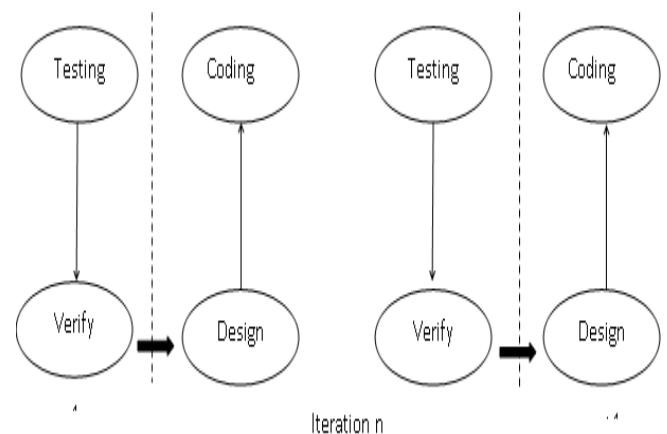


Figure 2 Iterative Model

The software is developed at first on a very small scale and all the other steps are followed one by one, which are considered in the

system. More features and domain are designed, coded, tested, and added in entire software. In every cycle, it produces software that is completely visible and also has more and more features and the abilities compared to the previous one.

When each iteration process ends, the management teams are able to undertake the risk management process and to prepare ready for the next set of iteration. As a single routine system consist of little portion of the entire software process, it is so simple to manage the entire development process but the problem is, it consumes more number of resources.

### 2.3 Spiral Model

It is the collaboration of both the iterative and one among the SDLC model that provides hybrid nature. It can also be seen like if the user choose any one of the SDLC model and mingle it with the concurring cyclic process called as iterative model.

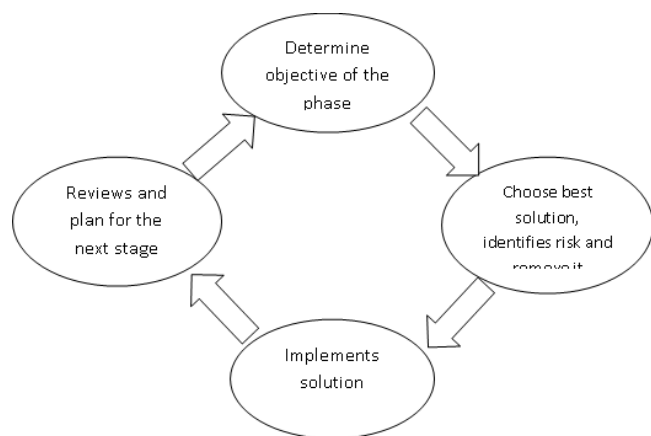


Figure 3 Spiral Model

This model has a major risk that may often exist un-noticed and unrecovered by some other strategy. This model starts with many deterministic objectives and limitations from the software at the beginning of each iteration.

### 2.4 V – Model

The main issues are to be noticed and corrected in the next stage after when the previous step is finished and there should not be any chance

to get back, if something is identified as wrong in the later stages. The V-Model persists the sort of testing present in software at each modes of procedure in the reverse order.

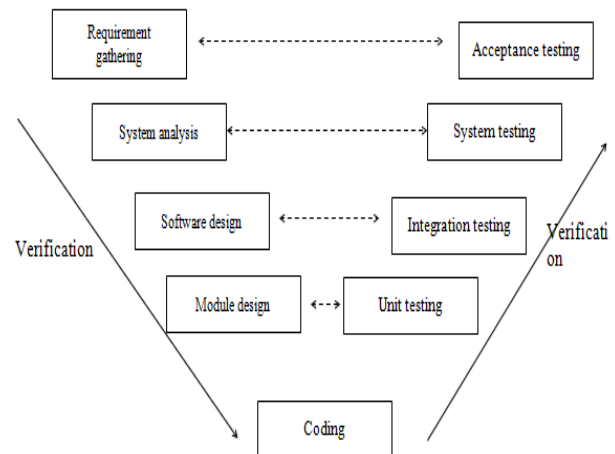


Figure 4 V- shaped Model

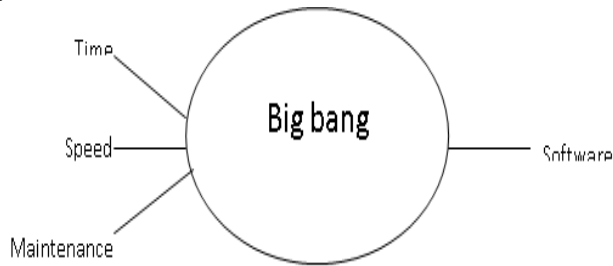
At every stages of this model, the plans and cases of the test were created in order to verify, analyze and validate the developed product related to the requirement condition of that particular stage. This is explained in an example, at the stage of collection of requirements, with which test team makes every test case or set in co-ordination to all the requirements present. The test cases from this stage verify the software against the product validity towards its requirements at this phase.

This ignites the both verification and validation process to go in a parallel manner. This V-shape model may called as verification and validation model respectively.

### 2.5 Big Bang Model

The simplest and effective design formation is found in this method. It needs a little planning of the project, lots or huge effort on programming and lots of expense. This model is documented around the way of big bang universe. Like how the scientists will tell that after big bang of galaxies, planets, sun and stars are evolved just as an unexpected event. As, if we put along the

lots of programming and costs, the best software product can be achieved from that.



*Figure 5 Big bang Model*

To design this model, only a small amount of product planning is required as a whole. It does not hope any process or method, as times of the customer access is not sure and also it is not clear about the needs and future scope. Thus, it said that the input requirements are generally arbitrary in nature.

### **3 : CONCLUSION**

As a result of growing maturity of software development it is no longer adequate that the software simply works, it must now meet the customer criteria hence In the above work various models related to software industry have been discussed.

### **REFERENCES**

- [1] K. Park, M. Ali and F. Chevalier, "A spiral process model of technological innovation in a developing country: The case of Samsung", *African Journal of Business Management*, vol. 5, no. 13, (2011), pp. 5162-5178.
- [2] P. Patil, "Quick Introduction to Agile Software Development", [www.indicthreads.com](http://www.indicthreads.com), (2008) June .
- [3] S. Easterbrook, "Software Lifecycles", University of Toronto Department of Computer Science, (2001). [5] I. Sommerville, "Software Engineering", Addison Wesley, 7th edition, (2004)
- [4] W. S. Humphrey, "A discipline for software engineering", Addison-Wesley Longman Publishing Co., Inc., (1995).
- [5] J. Feller and B. Fitzgerald, "Understanding open source software development", London: AddisonWesley, (2002)