

A Survey on Software Development Life Cycle Models

Shishir Jindal

Dr. Terry Feagin [CSCI 6530]

Research Paper for Fall-2016

University of Houston - Clear Lake

I. Abstract:

This survey paper provides a deeper insight to Software Development Life Cycle Models. It covers different properties of each model complimented with their advantages and disadvantages. This will help to have a better understanding in choosing a model for the development of a particular software. The flow of the paper deals with the introduction to Software Development Life Cycle and its phases followed by the study of Waterfall Model, Prototype Model, Incremental Model, Spiral Model, V shaped Model, RAD Model, Agile Model. The main idea of this paper is to have a comparative study of different models and make the best selection of the model for the development process. This results in a high quality and standardized software with minimum costs and optimized results.

Keywords: Software Development Life Cycle(SDLC), Prototype, Validation, Risk Analysis, Value

II. Introduction:

The purpose of this paper to provide brief understanding of Software Development Life Cycle (SDLC) and various Models that are being used for the Software Development. SDLC is a methodology which deals with constructing or fabricating systems, models and techniques used for the effective and predictable development of the software. It involves series of operation and phases which are employed to build software of required functionality using a definite set of standards within a specific deadline and budget. All software projects go through the phases of requirements gathering, business analysis, system design, implementation, and quality assurance testing ^[1]. Application of a particular life cycle model is very challenging and should be selected after all possible considerations as one model might be suitable in certain situations while another being perfect satisfying certain requirements and implementing functionalities. Application of life cycle models in development process ensures production of software which is standardized and has a better quality.

III. Phases of Software Development Life Cycle

Following are the list of significant activities that are involved in planning and development of software:

1. Requirement Analysis
2. Designing
3. Coding and Implementation

4. Testing
5. Maintenance

Requirement Analysis: In this phase information and specification of the product or the software is gathered. The developing team work in collaboration with the customer and lays down all the needs and conditions that are required to construct the system. It emphasizes on the functionalities that make up the entire system. This is achieved by analyzing and documenting the entire system from all the sides and a deep knowledge of the system is required before implementing it in order to have a good and correct set of the requirements.

Designing : This phase deals with providing a physical structure of the entire system or the project. It specifies the layout of entire software. Other factors such as project constraints, time and budget, skills of the developers are taken into consideration. After the requirement analysis the logical system design is converted into physical system design and this is facilitated by certain tools and techniques such as Data Flow Diagrams, Decision Table, Flowchart, Data Dictionary, Decision Tree, etc.

Coding and Implementation: This phase represents the development phase. In this the system is actually constructed by making use of certain programming languages. A software can even be constructed by making use of other softwares. The entire system is segregated into different modules and each module is worked on independently. This helps to avoid having dependency while implementation. The modules can then later on be submerged into one single system.

Testing: This phase deals with testing the system implementation against the set of requirements. This proceeds further by checking whether the implemented system matches up the specified requirements during requirement analysis phase and also it figures out the errors and faults or the bugs in the system. Once the bug is identified it is then carried back to the developers to fix it. After fixing of the bug it undergoes the verification process again to encounter any further defects. This is an iterative process until unless all the errors have been removed and can be taken to the next phase of the Software Development life Cycle.

Maintenance: This phase comes into picture after the software is being deployed at the client end and is functioning. It includes implementation of changes if software functionality and working undergoes any changes over a period of time or there happens to be a change in the requirement by the customer or the software has to be updated to a newer version etc. This is a very important step as this significantly contributes towards customer satisfaction.

IV. Models of SDLC

Over a period several models have been introduced for the Software Development Life Cycle. Selection of a model for a particular project is very important and should be chosen after considering all the requirements of the projects and all other factors too. This in turn is also crosschecked with the properties of the models and after a proper trade off a particular model is being chosen for the development of the software.

(1). Waterfall Model-

Waterfall Model was proposed by Royce in 1970 which is a linear sequential Software Development Life Cycle. It is the most basic model and has different phases like Requirement Analysis, Design, Coding and Implementation, Unit and System Testing, Maintenance. Execution of the subsequent steps can only take place after execution of previous steps. It is called as Waterfall Model because the workflow represents like that of a waterfall i.e. it is not possible to revert back to previous phases. This is typically a plan driven model in which the requirements of the software must be known upfront. This is suitable for small scale projects but as the complexity of the software increases its implication to the software development becomes difficult and it fails. Below is the simple diagrammatic representation of the waterfall model.

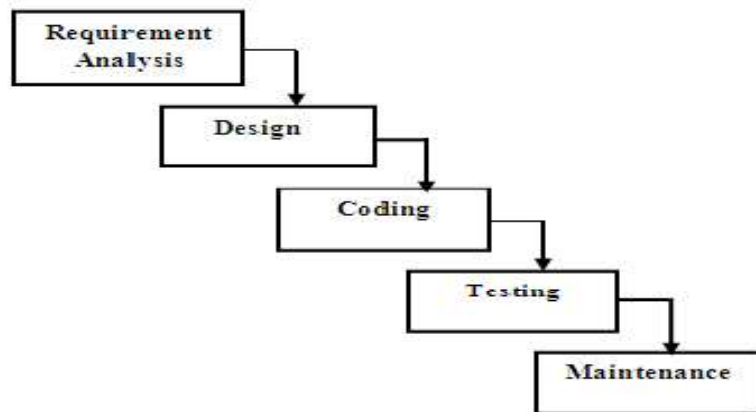


Fig 1: Waterfall Model [3]

Advantages of Waterfall Model :

- (i) It is easy to understand and use
- (ii) Error propagation prevented by Verification and Validation
- (iii) Each stage has well defined milestone

Disadvantages of Waterfall Model :

- (i) Requirements must be known upfront
- (ii) Progress of the development unknown
- (iii) Customer cannot demand any changes
- (iv) Going back to previous phase is not possible
- (v) Not applicable for large scale projects
- (vi) Life Cycle Model is too long
- (vii) Requirement of high quality planning

(2) Prototype Model:

This deals with approximating the system in the initial stages and creating a corresponding blueprint which is called as a prototype. It is an iterative process as after every prototype the system is studied again and as per the new changes in the requirements a new prototype is constructed. This process iterates until a final prototype is obtained. Final prototype is used for further implementation of the system. This model typically involves much of the customer

involvement as prototypes are made on the basis of the changes as suggested by the customer. Thus the customer works in close communication with the developers.

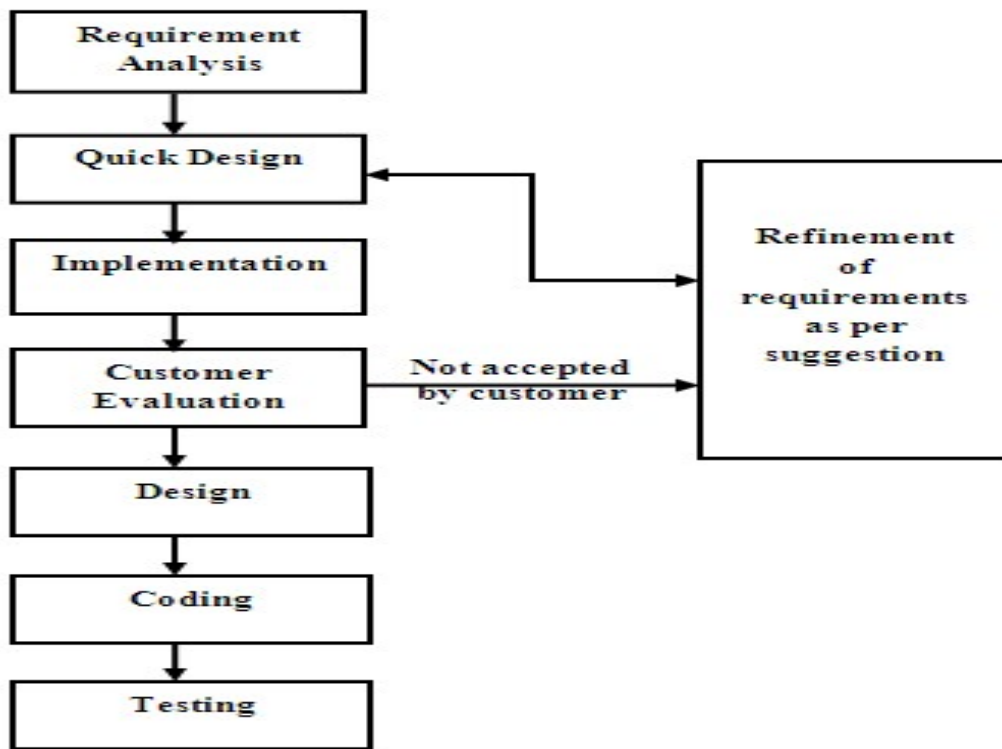


Fig 2: Prototype Model [3]

Advantages of the Prototype Model:

- (i) Active participation of the users in the process of development
- (ii) It gives a better understanding of the system to the users as it delivers working model of the system
- (iii) Detection of the error at the right time
- (iv) All the essential functionality can be identified
- (v) Validation of Requirements

Disadvantages of Prototype Model

- (i) This results in increase in the complexity of the system as continuous addition of functionality may result in going beyond the scope of the project
- (ii) If the requirements are implemented completely it results in software not getting used by the customer

(3) Incremental Model:

This is an incremental approach as functionalities are added in stages. It approaches with involved and closed communication of the customer with the developing team. In this model the requirements are prioritized and the system is constructed by adding functionality incrementally. The best part about this model is that a working mini project can be delivered to the customer. After each release the prototype comes with the added functionality. This helps the customer to

closely analyze the system, do some real work and demand further changes. These changes assume the form of functionalities which when added to the mini projects finally results into a final project or the system.

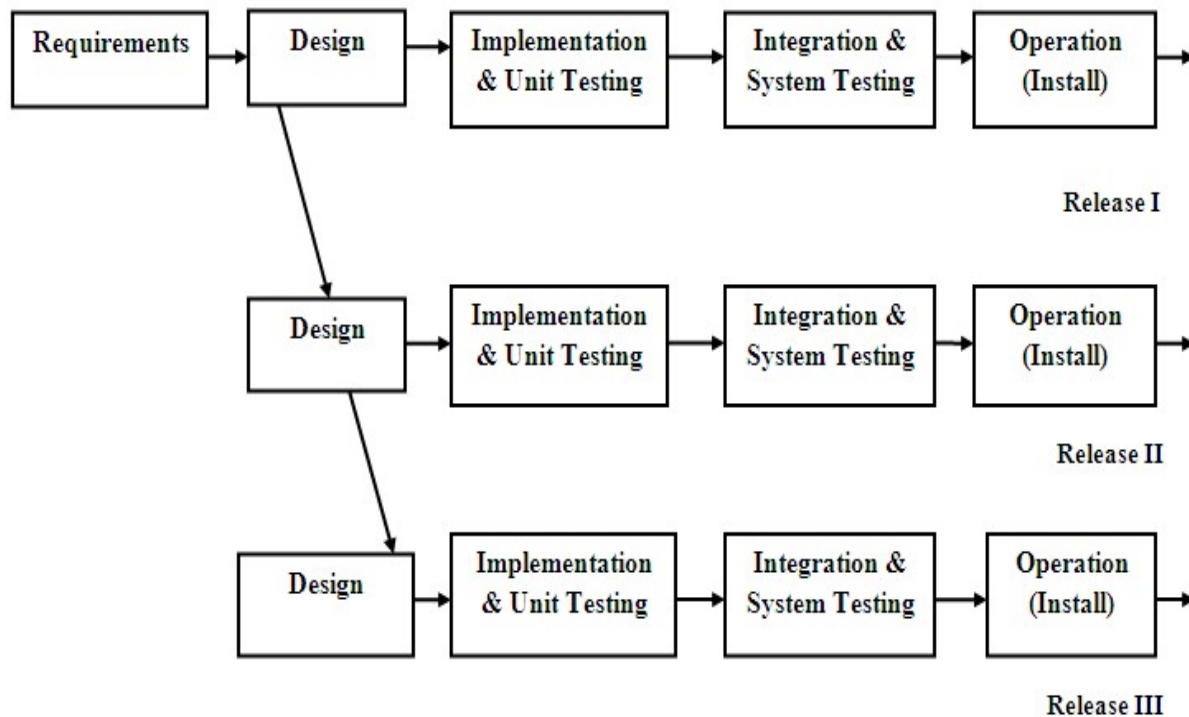


Fig 3: Incremental Model [3]

Advantages of Incremental Model:

- (i) It is easy to identify risks and overcome them during each iteration
- (ii) This model is very much responsive to customer demand of changes
- (iii) It has higher flexibility as it can easily respond to changing requirements and scope of the project

Disadvantages of Incremental Model:

- (i) It has strong dependency on good planning and design of the project
- (ii) It has much higher cost when compared to Waterfall Model
- (iii) It requires having a deeper insight of the entire system even before starting with the first stage

(4) Spiral Model:

This model is considered to be a hybrid of the models. The most important aspect of this model is that it takes into consideration a factor called Risk Analysis. The idea is that the entire model is cyclic in nature and is divided into four section namely: Planning, Risk Analysis, Development and Evaluation. Each of the individual phases represents a single quadrant. Planning phase is responsible for figuring out what are the necessary functionalities that are required to be included

in the system. Risk Analysis phase emphasized on identification of the risks and correspondingly providing alternate solution to it to optimize the development process. The importance of this phases lies under the principle that budget and time are two most critical aspects of the entire development process. If these attributes tend to be higher than expectation it decreases the value of the product.

Value = Amount paid by the customer - Cost(Budget and Time) incurred during development

The higher the positive value of the above equation the better the development process.

The radius of the spiral represents the cost and angular dimension represents the progress in process[4]. Development phase has same phases that of the Waterfall Model and deals with implantation of the project. Evaluation phase consists of customer providing the feedback and demanding some necessary changes in form of requirements in the project. At the end of the final cycle the end product is obtained.

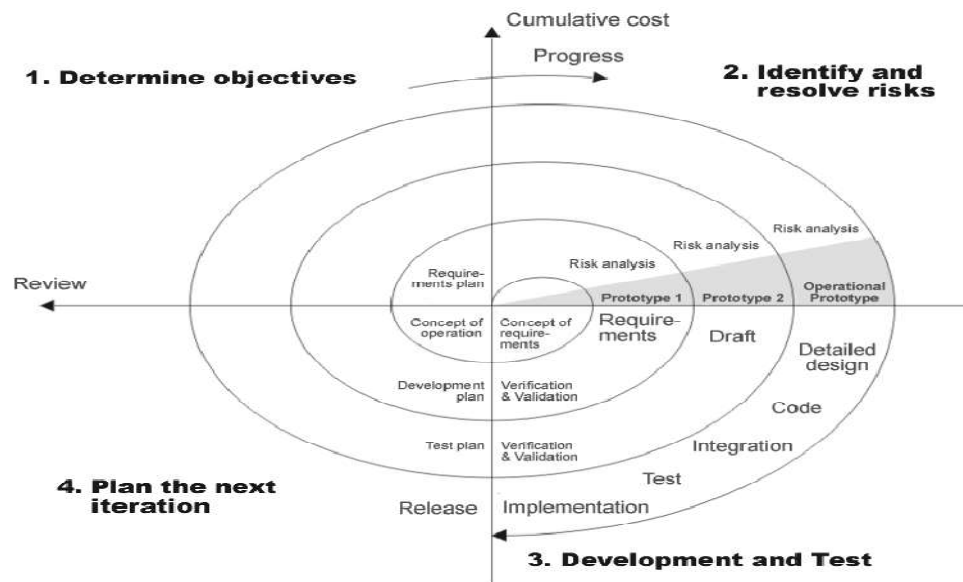


Fig 4: Spiral Model [2]

Advantages of Spiral Model:

- (i) It has high coverage of risk analysis
- (ii) It has high usage for large scale projects
- (iii) Chances of failure are significantly low
- (iv) It is flexible as the development process can be withdrawn at the end of any spiral but still it will result in a usable working product

Disadvantages of Spiral Model

- (i) It requires expertise in determination of risk

- (ii) It cannot be used for small scale projects as the cost that will be incurred on risk analysis might be greater than the total cost of the product
- (iii) Proper implementation of the project has a direct dependency on risk analysis.
- (iv) Estimation of the budget, time and others costs is difficult

(5) RAD Model:

It is an acronym for Rapid Application Development. It is characterized by high speed adaption of Waterfall Model. It is a very effective model in building quality assured projects in less time. It is used in scenarios where project has to be delivered in a very short time. In order to enable quick delivery of the product the set of requirements should be known in advance. It works by delivering small deliverables in form of mini projects to the customers. In this model also the customer works in close communication with the developing team. These deliverables are in working condition and customer after using them can demand further changes. This enables the processes of development to execute faster and thus the project delivered rapidly. It has four stages namely : Requirement planning, User description, Construction and Cut over. Requirement planning means identifying the set of functionalities that needs to be incorporated within the system. User Description means that the user collaborates with developers and help in formulating and deciding the design and specifications of the system. Development means actual implementation of system. Cutover deals with testing , training of the user, maintenance, etc.

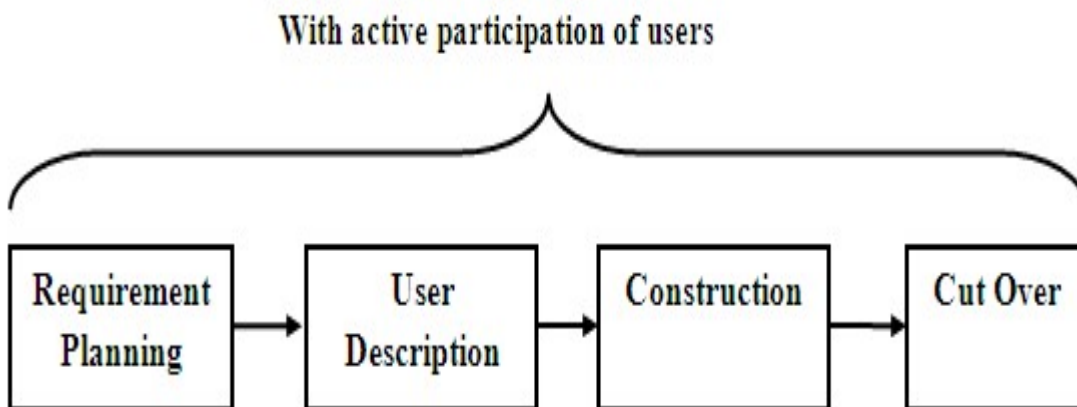


Fig 5: RAD Model [3]

Advantages of RAD Model:

- (i) Reusability of code
- (ii) It ensures user to interact with the developers
- (iii) The cost of the developed can easily be approximated
- (iv) The time required to develop reduces significantly
- (v) The probability of error occurrence is very low

Disadvantages of RAD Model:

- (i) It does focus on ensuring quality standards of the project

- (ii) It is not compatible with all application
- (iii) As soon intensity of risk occurrences increases it fails

(6) V shaped Model :

It is a type of Waterfall Model but it has strong emphasis on Verification and Validation. The testing of the product works in parallel with the phase of the development which implies that earlier tasks are verified later. It is a Sequential Model in which implication of a phase takes place only when previous phases have been implied. It does not have iterative phases as compared to Incremental Model and Prototype Model.

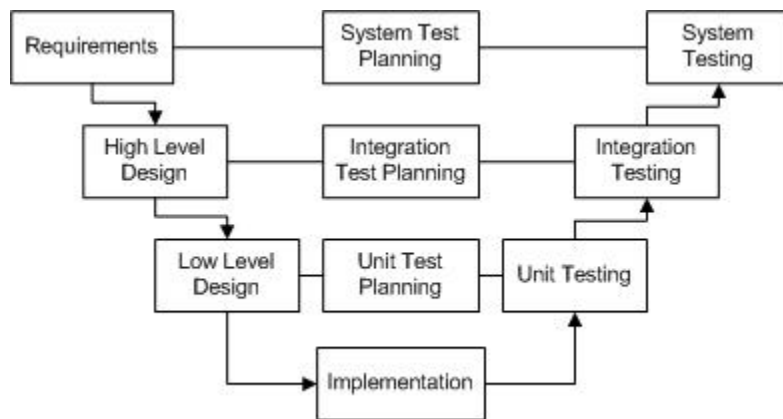


Fig 6: V shaped Model[3]

Advantages of V shaped Model:

- (i) It is easy to plan and understand
- (ii) Error can be identified and corrected earlier
- (iii) It saves a considerable amount of time as testing is done before coding so error detection becomes very easy and less time consuming
- (iv) Ensures that the errors do not get transferred to the subsequent steps
- (v) Its working is much better as compared to that of Waterfall Model

Disadvantages of Waterfall Model

- (i) It is a compact and rigid model and lesser flexibility
- (ii) It should only be used for small and medium sized projects when requirements are beforehand
- (iii) It can only be used when there is sufficient availability of technical resources and people who have an expertise in using them.
- (iv) It has high chances of risks and does not go through risk analysis
- (v) It does not provide prototypes to facilitate user to have a picture of the system being constructed

(7) Agile Model:

Agile in itself means able to move easily and quickly. Agile software model was developed by team agile in the year 2001. It results in quick delivery of the project. It works by using collaborative incremental and iterative approach leading to a high level of customer satisfaction. It involves cross training techniques and cooperative approach by effective communication between customer and the developer. Its working is based on team work which is implemented by the cross functional groups. There is flexibility in moving from one phase to that of another. This implies that if the developer is in design phase it can easily go to the requirement analysis step. This model is different from the other models in the sense that the earlier are plan driven and agile model has incremental planning approach.

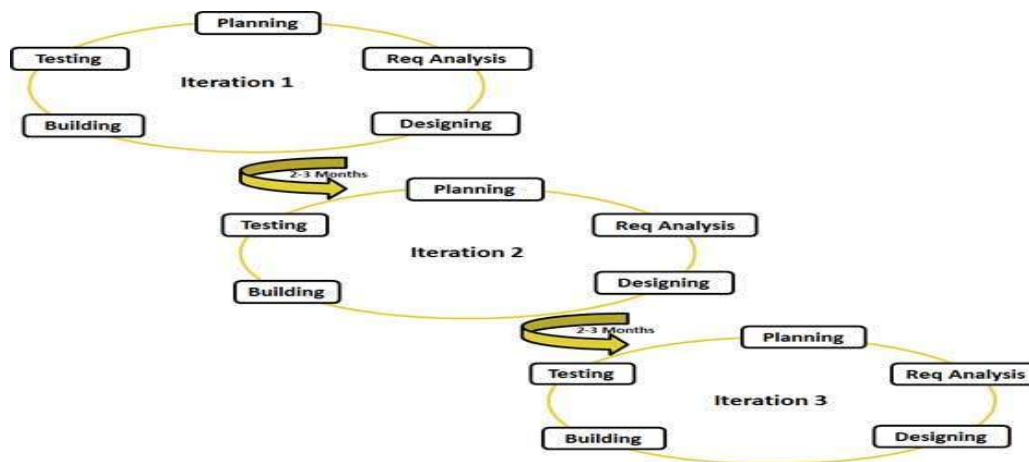


Fig 7: Agile Model [12]

Advantages of Agile Model:

- (i) It promotes team work and cross training
- (ii) Rapid development of functionality
- (iii) Requirement of the resources are minimum
- (iv) It can easily entertain changes in the requirements
- (v) It delivers partial working projects

Disadvantages of Agile Model

- (i) It is not applicable for complex systems
- (ii) It has higher risk of sustainability and extensibility
- (iii) Minimum documentation is produced
- (iv) It has high dependency on an individual as project cannot be transferred as it does not undergo documentation process

V. Comparative Study:

According to the properties of different models studied so far it can be tabulated to compare each model with that of another. This will help to identify which model would suit to develop a particular software.

Model/ Features	Waterfall	Prototype	RAD	Incremental	Spiral	Build & Fix	V-model
Well defined requirements	Yes	No	Yes	No	No	No	Yes
User involvement in all phases	Only at beginning	High	Only at beginning	Yes(Intermediate)	High	No	No
Risk analysis	Only at beginning	No Risk analysis	Low	No Risk analysis	Yes	No	Only at beginning
Overlapping phases	No overlapping	Yes	No	No	Yes	Yes	No
Implementation Time	Long	Quick	Quick	Long	Long	Depend upon project	Long
Cost	Low	High	Low	Low	Expensive	Low	Expensive
Incorporation of changes	Difficult	Easy	Easy	Easy	Easy	Difficult	Difficult
Simplicity	Simple	Simple	Simple	Intermediate	Intermediate	Simple	Intermediate
Flexibility	Rigid	Little Flexible	High	Less Flexible	Flexible	Flexible	Less flexible

Fig 8: A comparative study of SDLC Models [3]

VI. Conclusion

There are several software development models available out of these the best software must be selected. The primary goal of the software development process should be ensure that the software is developed quickly and it has all the requirements as listed by the customer. Using right model increases the productivity and functioning of the software. Application of these models also helps to have particular standards being followed throughout the project. This leads to the project getting easily transferred to different individuals and helps in better understanding of the entire system. Customer involvement results in rapid delivery of the project. Taking into consideration of all the aspects leads to a project that is quality assured. After development the user training is also very important to enable him to use the software efficiently and easily therefore maintenance department should regularly provide required services to the customer. Customer Satisfaction should be the main focus of the entire development process as this increases the value of the software.

VII. References

- [1] Klopper, R., Gruner, S., & Kourie, D. (2007), "Assessment of a framework to compare software development methodologies" Proceedings of the 2007 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries, 56-65. doi: 10.1145/1292491.1292498.
- [2] Apporva Mishra, Deepti Dubey, October 2013 "A Comparative Study of Different Software Development Life Cycle Models in Different Scenarios" ,
- [3] Prateek Sharma, Dhananjaya Singh , April 2015 "Comparative study of different SDLC models on different parameters",
- [4] What is Prototype Model – advantages and disadvantages and when to use it retrieved from <http://istqbexamcertification.com/what-is-prototype-model-advantages-disadvantages-and-when-to-use-it/>
- [5] What is Incremental Model – advantages and disadvantages and when to use it retrieved from <http://istqbexamcertification.com/what-is-incremental-model-advantages-disadvantages-and-when-to-use-it/>
- [6] Mishra A., Dubey D.,” *A Comparative Study of Different Software Development Life Cycle Models in Different Scenarios*”, IJARSMS, Volume 1,Page 64-69,October 2013.
- [7] Shubmeet Kaur, “A Review of Software Development Life Cycle Models”, International Journal of Advanced Research in Computer Science and Software Engineering”, Vol 5, Issue 11, November 2015
- [8] Abrahamsson P., Warsta J., Siponen M. and Ronkainen J., “New Directions on Agile Methods: A Comparative Analysis”, *Proceedings of the International Conference on Software Engineering*, May 3-5, 2003, Portland, Oregon, USA.
- [9] Nabil Mohammed Ali Munassar And A.Govardhan, “A Comparisonbetween Five Models of Software Engineering” *Ijcsi International Journal of Computer Science Issues, Vol7, Issue 5, Sep, 2010.*
- [10] 15A. 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, Issue10, 1988.*
- [11] Laura C. Rodriguez Martinez, Manuel Mora ,Francisco,J. Alvarez, “A Descriptive/Comparative Study of the Evolution of Process Models of Software Development Life Cycles”, Proceedings of the 2009 Mexican International Conference on Computer Science IEEE Computer Society Washington,DC, USA, 2009.

[12] SDLC Agile Model retrieved from
https://www.tutorialspoint.com/sdlc/sdlc_agile_model.htm