Comparative Review of Software DevelopmentLife Cycle Models

Hari Ji
Department of CSE
Rama University, Kanpur, India
jihari@gmail.com

Anurag Tiwari
Department of CSE
Rama University, Kanpur, India
anurag.cse05@gmail.com

***Abstract*- They compare the different software development models that have been carried out in this paper, the mathematical approach cannot be made with certainty the value or physical in the present age of the software system models are incomplete. We develop software and towards the creation of new or improved systems need to follow a structured approach that emphasized the concept of the system life cycle model came into existence. Many models etc waterfall, iterative model, prototype model, spiral model and RAD were taking here.**

Keywords- Software Development Process, Software Development Life Cycle, Activities involved in SDLC models, Comparative analysis of models.

I. INTRODUCTION

A software development process is known as software development life cycle (SDLC) these models use to develop the projects. Because the development of software product follow the software development norms or rule. A software development process, also known as a software development life cycle (SDLC), is a structure imposed on the development of a software product. It is often considered as a subset of system development life cycle. In software development different phase these phases can be considered as life model. There are several models for such processes, each describing approaches to a variety of activities that take place during the process. ISO 12207 is an ISO standard for describing the method of selecting, implementing and monitoring the life cycle for software. [1] Any software development process is divided into several logical stages that allow a software development company to organize its work efficiently in order to build a software product of the required functionality within a specific time frame and budget. Rodriguez-Martinez et al. [2] focused on life cycles frameworks models and detailed software Development life cycle process and report specification as well as hardness, general, specific, and unique in terms of the elements of a plausible explanation of their development and agility attributes permit a comparative study of the software development life cycle results. For this, a conceptual research approach and a software lifecycle process meta-model is used. Jovanovic D et al. [2] as compared to the basic principles and software development model is presented. The first part and the second part of the presentation of the development model of software development models introduces a practical approach to implementing. Finally, developing PC applications [3] in terms of the problem of determining the appropriate software development model. Davies AM Et al. [4] describes the potential effects of a life-cycle model to help researchers a tool for software engineering life-cycle as an alternative to analyze the similarities and differences between models as a basis and software to help that can serve as a means of providing a framework for practitioners in a particular project or a particular application to use to decide on an appropriate life-cycle model. Sharma, B. et al. [5] made a comparative analysis of software process improvement model. Software process improvement of software development life cycle is recognized as an important part. The study also Capability Maturity Model, ISO etc. provided emulates existing models as well as their significance and drawbacks of each model analyzes.

Maglyas A. et al. [6] the size and complexity of software development projects are still at the same time increases in proportion of successful projects that are significantly lower described. The purpose of this study predicted Standish Group and McConnell model to compare two existing models and to determine their strengths and weaknesses and the results of the Standish Group, a project to address the problems that have a tendency to overestimate walks. McConnell predicted very well managed projects, but the percentage of failed projects underestimates. Cries et al. [7] provides a brief review of traditional SDLCs, they use many of the traditional software development model and often for analysis and design of software applications that are considered reasonable and disciplined approach is related to. Osborne et al. [8] discussed techniques and traditional SDLC phases over time these methods, the requirements for defining requirements, coding and testing of a system designed to meet that are inherent in how the development cycle. A software development life cycle models break down and these activities in different activities are held in the software development effort that specifies how. In response to the traditional methods of software development, new light has appeared methods [9].

A software development life cycle model that will produce the desired product ordering, performing activities in a manner that satisfies the relationship between a set of activities with a command relationship. SDLC process of developing a model "is an abstract representation of Fig. 1,". The goal of a software development effort is to produce high quality software. Development process so that the sequence of activities that will produce such software. A software development life cycle models break down and these activities in different activities are held in the software development effort that specifies how. In response to the traditional methods of software development, new lightweight methods have appeared. [8]

II. ACTIVITIES INVOLVED IN SDLC MODEL

Problem solving in software consists of these activities:

- Understanding the problem
- Deciding a plan for a solution
- Design of the problem
- Coding the planned solution
- Testing the actual program

For large systems, each step we cannot change because the software is so big you need the information necessary for software development is a process and method are very important and complex. Also, basic activities cannot be handled by themselves and at each step it must be broken into smaller steps that can be so large. For example, a large software system design is always specified logic components, where a detailed design of the system components, specifying only the start of a very high level design, broken into many different design stages is there. Basic activities or steps to be performed to develop a software system

- Determination of System's Requirements
- Design of system
- Development (coding) of software
- System Testing

After the development is complete except for the activities performed during the development of software, some activities are performed. System installed on the customer's computer system and then it is concerned with testing, which is an installation phase, there is often. After the software has been developed to maintain an activity that is initiated. The software components of the "wear out" and need to be replaced because not need to be, but often they are discovered must be removed after the rest of the system because some residual errors. So, is indispensable for the maintenance of software systems.

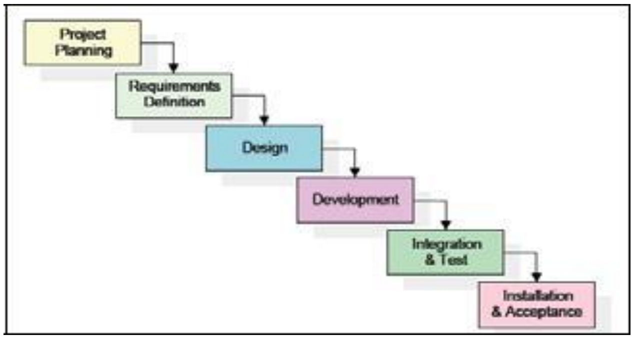

Fig.1. SDLC Model

The requirement in commercial software development takes place before the analysis are performed certain activities. The feasibility analysis is used to analyze the feasibility phase of the project, and a business proposal and cost estimate for the project with a very general plan is put forward. Accepted business proposal or contract is awarded, the development activities need to start with the beginning of the analysis.

III. COMPARATIVE ANALYSIS

We have V-shaped waterfall, CMM, prototyping, incremental, spiral and red model will discuss different models. The requirements are very well understood, which works well for small projects. During the life cycle of the V-shaped model for the development of test plans is high chance of success on the waterfall model. The requirements model easily understood.CMM alone.CMM usually present in different parts of the stovepipes and provides an opportunity to eliminate barriers to use than other process improvement products more detailed coverage of the product life cycle are provided, where the organization works well for small projects that usually are not addressed by the other process improvement model.

Instead of actual prototype model making software makes more effort to focus on documentation. Prototype models can easily change their requirement provides flexibility for the user. Thus, the actual software may be issued in advance. Users

TABLE I. COMPARATIVE ANALYSIS

Model/Feature	Waterfall	V-shape	CMM	Prototype	Incremental	Spiral	RAD
Requirement Specification	Beginning	Beginning	At second Level	Frequently Changed	Beginning	Beginning	Time Bound
Understanding Requirements	Well Understood	Easily	Easily	Partial Understood	Well Understood	Well Understood	Easily
Cost	Low	Expensive	High	High	Low	Expensive	Low
Guarantee of Success	Low	High	High	Good	High	High	Good
Resource Control	Yes	Yes	Yes	No	Yes	Yes	Yes
Cost control	Yes	Yes	Vary to Project	No	No	Yes	Yes
Simplicity	Simple	Intermediate	Intermediate	Simple	Intermediate	Intermediate	Simple
Risk	High	Low	Varies at Level	Low	Easy Manage	Low	Low
Experts Involvement	High	Medium	Varies at Level	Medium	High	High	Medium
Changes Incorporated	Difficult	Difficult	Medium	Easy	Easy	Easy	Easy
Risk Analysis	Only at Beginning	Yes	Yes	No Risk	No Risk	Yes	Low
User Involvement	Only at Beginning	At The Beginning	Only at Beginning	All Time	Intermediate	High	Initial Level
Overlapping	No	No	No	Yes	No	Yes	No
Flexibility	Rigid	Little Flexible	Highly Flexible	Highly Flexible	Less	Flexible	High
Maintenance	Least	Least	Typical	Routine Maintenance	Partial	Typical	Easy
Integrity & Security	Vital	Limited	Limited	Weak	Robust	High	Vital
Reusability	Limited	Medium	Yes	Weak	Yes	Yes	Some Extent
Interface	Minimal	Minimal	Crucial	Crucial	Crucial	Crucial	Minimal
Documentation & Testing	Vital	Yes	Yes	Weak	Yes	Yes	Limited
Time Frame	Long	Project size	Quite Long	Short	Very Long	Long	Short
Feasibility	High	Medium	High	Medium	Medium	High	Low

always interact with the developer. Hence the need for the likely to meet, feel and performance.Incremental model is at the heart of a cyclical process of software development. The model developed in the second phase after the first phase of development. It starts with an initial plan and the cyclical relationship between ends with deployment. During a short visit easier to test and debug. The first step is incremented to the next stage of debugging easy. Risky pieces are identified and controlled during his visit, as easy to manage risks. Spiral model of risk analysis is necessary as the high amount of satellite launch, where is good for large and mission-critical projects.It means looking at the product development cycle, including red as users quickly to changes in the red model is flexible and

product is in relation to the final product will of the user is more adaptable. In the short time they get their product is good for the customer provides that within 90 days. It also increases the likelihood of early user participation and user community acceptance of the project is to realize an overall reduction in risk. The spiral model is used for large projects. In-phase spiral model, top -down and attempts to combine the advantages of the above concepts and prototypes for the design of a software development process combing elements. This is a meta-model that can be used by other models is a model. during the development process, as well as on providing the cycle re-use capabilities for improved productivity through the smooth working and there is a new prototype, the more customer involvement is then received by the customer again and again revaluated is used for each part of each time, progress through the same sequence of steps, including the work of the table for a project cost, time and manpower required to control accurately. Errors in the early stages of project development are eliminated only.

IV. CONCLUSION

Depending on the needs of the software for any software used in the development of many of the SDLC models and also depend on the particular software development organization. Waterfall model that we believe we have no knowledge of the user when we all need to use. It is very useful for the organization. All these different software development model has its advantages and disadvantages. Contemporary commercial software development world, the company's uses of mixed models, some organizations use these models. Time is very important in the development of software. Is a delay in the development phase, can be taken up by the market competitive. These models are very useful for developing software and gaps in various stages of developing software to provide different functionality. Next we use these models to verify these models can develop mathematical approach.

REFERENCES

[1] Comparative Analysis of Software Development Life Cycle Models ISSN: 0976-8491 (Online) ISSN: 2229-4333Decb. 2011.

[2] 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.

[3] Jovanovich, D., Dogsa, T.,“Comparison of software development models,” Proceedings of the 7th International Conference on, 11-13 June 2003, ConTEL 2003, pp. 587-592.

[4] 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

[5] Sharma, B.; Sharma. N, “Software Process Improvement: A Comparative Analysis of SPI models”, Emerging Trends in Engineering and Technology (ICETET), 2009 2nd International Conference on,16-18, 2009, pp. 1019-1024

[6] Maglyas, A.; Nikula, U.; Smolander, K.,“Comparison of two models of success prediction in software development projects”, Software Engineering Conference (CEE-SECR), 2010 6th Central and Eastern European on 13-15 Oct. 2010, pp. 43-49.

[7] Osborn, C. SDLC, JAD, RAD,"Center for Information Management Studies", 2001.

[8] Rothi, J.,Yen, D, "System Analysis and Design in End User Developed Applications", Journal of Information Systems Education, 1989.

[9] Fowler, M. (2000), "Put Your Process on a Diet", SoftwareDevelopment".