Imperial Journal of Interdisciplinary Research (IJIR)
Vol-2, Issue-6, 2016
ISSN: 2454-1362, http://www.onlinejournal.in

Waterfall vs. Prototype: Comparative Study of SDLC

Aayushi Saxena[1]&Priya Upadhyay[2]

[1]Lecturer in C.S.E dept. Vivekananda college of Tech. &mgmtAligarh(UP).
[2]Student in C.S.E dept. Vivekananda college of Tech. &mgmt Aligarh(UP).

Abstract*: A major challenge in any engineering endeavor is taking a poorly organized, uncertain, inconsistent, incomplete, and over generalized requirements specification and turning it into a well-structured design. Software development is highly complex & interesting field with so many variables having lots of impact on system. There are various models for development of software. This paper aims to describe the impact and comparison of various methodology ,to decide which model is more suitable & appropriate by analyzing their software matrices, development time, complexities, return on investment with minimal initial capital expenses, development cost, as we know success of a software depends upon which process model is being used.*

Keywords*: SDLC, Comparative analysis of SDLC models, waterfall, prototype.*

I.INTRODUCTION:

A computer plays a very vital role in our everyday life.It has become indispensable as it is used in every field of life such as industry, medicines,education,agriculture,commerce etc.Every organization either small or large is dependent on information technology.So to make an organization under the role of computer world there is a need of different software's which are used to make the work of an organization as convenient and efficient. To develop the software's there is a need of software development life cycle.A proper software life cycle model can help an organization not only in building a software product but it also serves as basis in planning, organizing, staffing, coordinating and directing. According to IEEE standard terminology, software development life cycle is defined as "The period of time that starts when a software product is no longer available for use".There are various phases of SDLC.

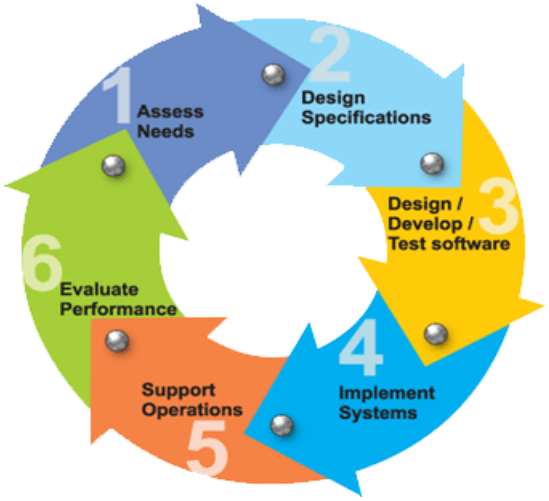

Fig1: system development life cycle

II.WATERFALL MODEL:

Waterfall model was proposed by Royce in 1970 which is a linear sequential software development life cycle (SDLC) model.This model is named "Waterfall" because its diagrammatic representation looks like a cascade

(flow) of Waterfall. This is also known as classical lifecycle model.It includes system, requirements, initial and detailed design, implementation, testing, operations,and maintenance.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 occurring at the end of most phases before beginning the next phase.

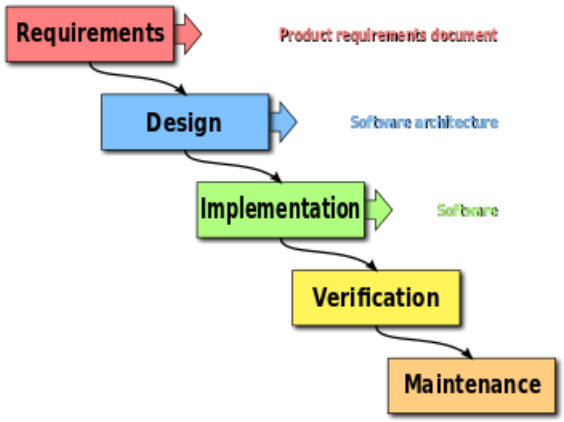

Fig2:Waterfall model

PROS

- Each stage has well defined deliverable or milestone.
- It is simple to use and understand.
- Each phase is completed in specified period of time after that it moves to next phase.
- As its linear model, it's easy to implement.
- The amount of resources required to implement this model are minimal.

CONS:

- The biggest disadvantage of the waterfall model is one of its greatest advantages. You cannot go back a step; if the design phase has gone wrong, things can get very complicated in the implementation phase.

- Often, the client is not very clear of what he exactly wants from the software. Any changes that he mentions in between may cause a lot of confusion.

- Small changes or errors that arise in the completed software may cause a lot of problems .
- Another major disadvantage of the waterfall model is that, until the final stage of the development cycle is Complete, a working model of the software does not lie in the hands of the client.

III.PROTOTYPE MODEL

A prototype model is a system development which does not require freezing the requirements.In this model,there is a first step to create a rough design or prototype, test it and rework as necessary until an acceptable prototype is finally achieved by which a complete product can be developed.Hence it is a trial and error process that takes place between the developer and the user. this type of system development method is employed when it is difficult to obtain the exact requirements from the customer.In this model, a prototype acts as a sample to test the process. By prototyping, the software designers and implementers can get a valuable feedback from users early in the project.It is an attractive idea for complicated and large systems.

Fig3:Prototype model

PROS

- Users can actively participate in development.
- Errors can be detected easily.
- Users feedback
- Better understanding of system being developed even for non-I.T literate people.

IV.COMPARISION OF WATERFALL & PROTOTYPE MODEL

Comparison between both SDLC models with necessary features so that is become easy to choose which model is suitable according to requirement.

Features	**Waterfall**	**Prototype**
Requirement specification	Beginning	Frequently changed
Cost	Low	high
Simplicity	Simple	Simple
Risk Involvement	High	No risk involvement
User involvement	Only at beginning	High
Flexibility	Rigid	Little flexible
Maintenance	Least	High
Patience level	High	Low
Documentation	high	Less documentation
Suitability	For simple system	For complex system

V. CONCLUSION

A discussion on various SDLC models has been obtained in this paper. Although many development models exist, this paper discusses different models out of those and the comparison includes the advantages and disadvantages of two models which can help to select specific model at specific situations depending on customer requirement and including business requirements. It gives a comparison of prototype and waterfall models, and also describes the benefits of prototype over waterfall methodology. Prototype model used to develop online systems for transaction processing. Since significantly reduce rework and lead to the creation of working model in lower capital cost.

REFRENCES

[1] Steve Easterbrook, "Software Lifecycles", University of Toronto Department of Computer Science, 2001.

[2] Rashmi Popli1, Anita and Nareshchauhan "Mapping of Traditional Software Development Methods to Agile Methodologys".

[3]M, Hue, 1. Varner, L. Zhu, and M, A. Babar, "Software quality and Agile Methods," Pros,28th Annual International Computer Software and Applications Conference (COMPSAC'04), pp., 520-525,2004.

[4] Guy Want, "Drowning in the Waterfall, The Benefits of Agile versus the Predominance of Waterfall " Software Engineering CS 390:", October 29, 2008.

[5].Frye, Colleen. "Agile practitioners face challenges, but see process improvements." Search Software Quality. 27 June 2008. 20 Oct 2008

[6] Lillian Doric and Elias Niemelae "A Survey on Software Architecture Analysis Methods", IEEE Transaction on Software engineering, VOL. 28, NO. 7, JULY 2002

[7] Jeffrey A. Livermore Walsh College, "Factors that Impact Implementing an Agile Software Development Methodology".

[8] Eduardo Malaga Chicano, "A comparative study if iterative prototyping vs. waterfall model processed by small and medium sized projects by system engineering", 1996.

[9] Nabil Mohammed Ali Munassar, Dr. A. Govardhan, "Comparison study between Traditional and Object-5th Malaysian Conference in Software Engineering",2011.