ISSN No. 0976-5697

Volume 2, No. 6, Nov-Dec 2011

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Study of Software Process Model Selection

Er.Neha Budhija*
M.Tech(CSE)
IGCE, Abhipur,
Mohali,Punjab,India
nbudhija@gmail.com

Er. Satinder Pal Ahuja
Associate Professor & HOD (CSE&IT),
Indo Global College of Engineering
Punjab,India
asatinderpal@gmail.com

Abstract- In the software field, a large number of projects fail and billions of dollars are spent on failed software projects. Nowadays it has been widely accepted that the quality of software highly depends on the process that is carried out in an organization. No exact system was found, which could guide Software Engineers for selection of a proper process model during software development. Lacks of poor selection process of software development life cycle (SDLC) models is some of the top reason of such failure. By selecting right software process model a better and high quality product can be found within budget and time. In this paper, an approach is proposed to select an appropriate SDLC model based on different project characteristic categories. In this paper, a comparison approach of SDLC process is introduced, which is based on project characteristic categories and then categories are classified. Paper is based upon better selection process of SDLC models.

Keywords: **-** Software Process Models, Categories of problems, SDLC Comparison tables

I. INTRODUCTION

This research work presents the selection of an appropriate software process model. The main goal of the research work is to guide the software engineers for decision making for selection and evolution of software process model [1]. Software process model is a representation of a software process. Software process model is an approach or method or both by which software process model efficiently selected depends upon the given requirement and give better result rather than a normal selection process. It aims to manage and transform the user need into a software product that meets their needs.

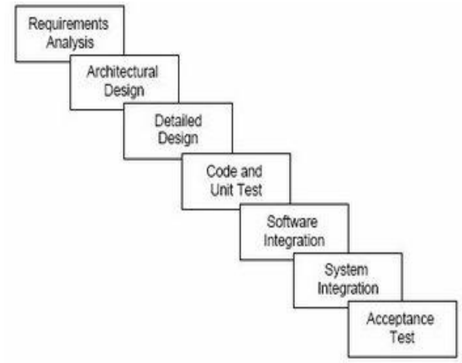
The requirement consists of questions related to the things which are requested by the user of the project. Improving the quality and reducing the cost of products are fundamental goals of software process model selection. As the qualities are determined by the process to satisfy the objectives of quality improvement and cost reduction, the software process must be improved. Thus the enforcement of the process model can directly contribute to the improvement of the software quality.

The process must be improved on the basis of past experience on software projects. The activity of analyzing and improving the process is largely done in process development component of the software process. From time to time so many software process models have been developed in order to maintain reliability and quality of software. Our research work focus on comparison of software process model for software application.

II. SOFTWARE PROCESS MODELS

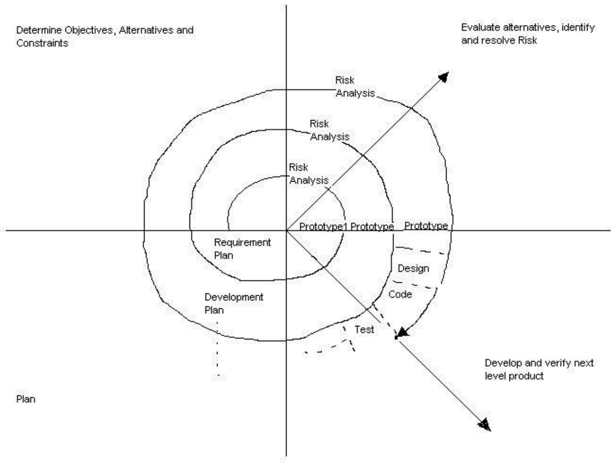
Software process models are defined only in terms of requirements analysis phase of each model.

A. Water Fall Model:



Waterfall Model is a poor choice for software development projects where requirements are not well-known or understood by the development team. It might not a good model for complex project or projects that take more than a few months to complete [2].

B. Spiral Model:


Determine Objectives, Alternatives and Constraints
Evaluate alternatives, identify and resolve Risk
Risk Analysis
Risk Analysis
Risk Analysis
Prototype1 Prototype Prototype
Requirement Plan
Design
Code
Development Plan
Test
Develop and verify next level product
Plan

In response to the weaknesses and failures of the Waterfall SDLC Model, many new models were developed that add some form of iteration to the software development process. In the Spiral SDLC Model as in figure 2 , the development team starts with a small set of requirements and goes through each development phase (except

Installation and Maintenance) for those set of requirements. Based on lesson learned from the initial iteration, the development team adds functionality for additional requirements in ever-increasing "spirals" until the application is ready for the Installation and Maintenance phase [6][7].

C. RAD Model:

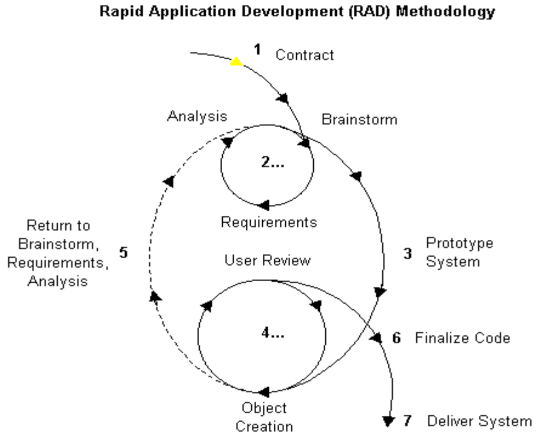


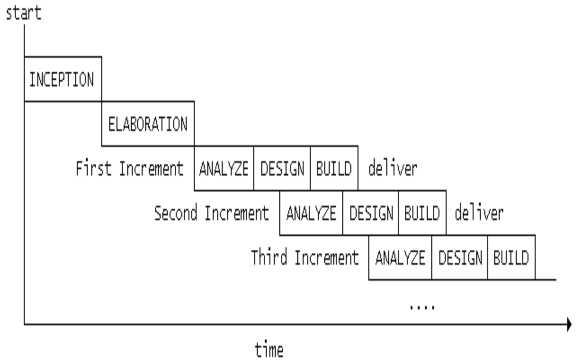
FIGURE: 1. WATER FALL MODEL [5].

RAD is a linear sequential software development process model that emphasis an extremely short development cycle using a component based construction approach. If the requirements are well understood and defines, and the project scope is constraint, the RAD process enables a development team to create a fully functional system with in very short time period [6].

This model assumes the requirements to remain static during the life of the project, so there is little or no chance of incorporating new changes to the software once work begins. If changes are tried to be incorporated it leads to more confusion and further delays.

The major weakness of the Waterfall Model as in figure is that it does not allow for much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage. Often the product that is implemented at the end of the process is obsolete as it goes into production. The

D. Incremental Model:



The incremental model figure 4 is a method of software development where the model is designed, implemented and tested incrementally (a little more is added each time) until the product is finished. It involves both development and maintenance. The product is defined as finished when it satisfies all of its requirements. This model combines the elements of the waterfall model with the iterative philosophy of prototyping. That is basic requirements are addressed, but many supplementary features (some known, others unknown) remain undelivered. The core product is used by the customer (or undergoes detailed review). As a result of use and/or evaluation, a plan is developed for the next increment. The plan addresses the modification of the core product to better meet the needs of the customer and the delivery of additional features and functionality [7].

3. Problem arises in software process

A lot of problem may become the reason of unsuccessful project.

Lack of management, Lack of well communication, Lack of well communication between costumer and organization, Technical problem and lack of resources, Lack of man power, not provided a good training.

III. CATEGORIES OF PROBLEM

	Cause of problem	Category out of 5 scale
1	Lack of management	4
2	Lack of well communication	3
3	Lack of well communication between costumer and organization	3
4	Technical problem and lack of resources	4
5	Lack of man power	3
6	Not provided a good training	3

IV. SDLC COMPARISON TABLES

Project characteristic is measure in 0-10 rating.

Comparison tables are design on three project Characteristic categories.

a. Project Team
b. User Community
c. Project type and Risk

A. Project Team:

Whenever possible, it is best to select the people for the project team before selecting any SDLC process model. The characteristics of this team table 1 are important in the selection process they are responsible for successful completion of the cycle, and they can assist in selection process.

Characteristics of the project team members:-

a. ***New to Problem Domain:*** - are the majority of team members new to the problem domain for the project?

New to the technology domain: - are the majority of the team members new to the technology domain for the project?

b. ***New to Tools to be Used:*** - are the majority of team member new to the tools to be used on the project?

Any training available: - is there training available for the project team, if required?

c. ***Comfortable with Structure:*** - is the team more comfortable with structure than flexibility?

d. ***Closely Track by Manager:*** - will the project manager closely track the team's progress?

Table: 1. Comparison based on project team

	Water fall	Spiral	Rad	Incremental
New to problem domain	1	9	1	3
New to technology domain	8	9	1	8
New to tools to be used	7	8	1	2
Any training available	2	1	8	9
Comfortable with structure	8	1	2	9
Closely track by manager	8	9	2	9

B. User Community:

The early project phase can provide a good understanding of the user the user community table 2 and expected relationship with the project team for duration of the project. This understanding will assist you in selecting the appropriate model because some models are dependent on higher user involvement and understanding the project.

Characteristics of the user community:-

a. Availability of user representative restricted or limited: - will the availability of the user reprehensive be restricted or limited during the life cycle?
b. User representative new to the system definition:-are the user representatives new to the system definition?
c. User representative expert in problem domain:- are the user representatives expert in problem domain?
d. User representative want to involve in SDLC:-does the user want to involve in all phases of the life cycle?

Table: - 2. Comparison Based On User Community

	Water fall	Spiral	Rad	Incremental
Availability of user representative restricted or limited	9	2	2	7
Expert in problem domain	2	9	2	8
Want to track the project process	7	8	1	2
Want to involve in SDLC	2	9	2	8

C. Project Type and Risk:

Examine the type of project and risk table 3 that has been identified to this point in the planning phase. Some models are designed to accommodate high-risk management, while others are not. The selection of a model that accommodates risk management does not mean that you do not have to create an action plan to minimize the risk identified. The model s\imply provides a framework within which this action plan can be discussed and executed.

Characteristics of project type and risk:-

a. ***Integration Project:*** - is the project a system integration project?
b. ***Enhancement to an Existing System:-*** is the project an enhancement to an existing available project?
c. ***The Funding for Project: -*** is the funding for the project expected to be stable through-out the life cycle?
d. ***Project Reliability:*** - Is the project high reliability a must?

Table: 3.Comparison Based On Project Type And Risk

	Water fall	Spiral	Rad	Incremental
System integration project	2	8	7	9
Enhancement to an existing project	2	2	9	8
High reliability is must	7	9	8	2

V. CONCLUSION

Based on observation, comparison and experience tables 4, 5, 6 are prepared and the steps in best life cycle selection are these:

a. Being familiar with various models.
b. Review and analyze the types of work performed like development, enhancement, and maintenance.
c. Review the life cycle approach to standards required for your organization, your customer, or the type of project- ISO, IEEE, and so on.
d. Identify a set of phase and phase activates.
e. Evaluate the effectiveness of the life cycle framework, and implement improvements where needed.

Table: 4. Suggested model base on team property

S.N.	Project team members	Suggested Model
1.	New to problem domain	Spiral
2.	New to the technology Domain	Spiral
3.	New to tools to be used	Spiral
4.	Any training available	Incremental
5.	Comfortable with structure	Water fall
6.	Closely track by manager	Spiral

Table: 5.Suggested Model Based On Use Community

S.N.	**User Communicate**	**Suggested Model**
1.	Availability of user representative restricted or Limited	Water fall
2.	User representative new to the system definition	Spiral
3.	User representative expert in problem domain	RAD
4.	User representative want involve in SDLC	RAD
5.	User representative want to track project progress	Spiral

Table: 6. Suggested model based on project type

S.N.	Project type and risk	Suggested Model
1.	Integration project	Incremental
2.	Enhancement to an existing System	RAD
3.	The funding for project stable	

4.	Project reliability must	Spiral

VI. REFERENCES

[1]. Abdur Rashid Khan, Zia Ur Rehman, Hafeez Ullah Amin, Knowledge-Based System Modeling for Software Process Model Selection (International Journal of Advanced Computer Science and Applications (IJACSA, Vol. 2, No.2, February 2011)

[2]. Manish Sharma,A Survey of project scenario impact in SDLC models selection process (International Journal of Scientific & Engineering Research Volume 2, Issue 7, July-2011)

[3]. RajPal Garg1, Kapil Sharma2, Member IEEE, Rajive Kumar3, R. K. Garg 2010 ,Performance Analysis of Software Reliability Models using Matrix Method

[4]. Devesh Kumar Srivastava, Durg Singh Chauhan, Square Model- A Software Process Model for software System (International Journal of Computer Applications (0975 – 8887) Volume 13– No.7, January 2011)

[5]. Ian Sommerville, “Software Engineering”, 8th Edition, 2006, pp. 89.

[6]. R.S. Pressman, “Software Engineering, A Practitioner’s Approach”,5th ed. New York: McGraw-Hill, 2001, pp. 26.

[7]. B.W. Boehm, “A Spiral Model for Software Development andEnhancement”, IEEE, IEEE Computer Society, vol. 21, issue 5, May 1988, pp. 61 – 72.

[8]. W.W. Royce, “Managing the Development of Large Software Systems: Concepts and Techniques”, IEEE, IEEE Computer Society, August 1970, pp. 1-9. Interscience, 2007, pp. 32.

[9]. Dinesh Kumar, Saroj Hiranwal” Performance Enhancement of Software Process Models” 2010 2nd International Conference on Software Technology and Engineering (ICSTE).

[10]. Robert H. Martin, “A Comparison of Software Process Modelling Techniques”.