

A Survey of project scenario impact in SDLC models selection process

Manish Sharma

Abstract— In the software industry, a large number of projects fail and billions of dollars are spent on failed software projects. 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 described about comparison tables of SDLC models, and better selection process of SDLC models.

Index Terms- Process model, Project team, Project type, Project risk, RAD model, SDLC, Spiral Model, Water fall model.

1. INTRODUCTION

Software Process Model is an abstract representation of a software process[1]. Each process model represents a process from particular perspective, and thus provides only partial information about that process [1]. Software process selection is an approach or method or both by which software process model efficiently selected depends upon the given requirements and give better result rather than a normal selection process. The requirements consist of questions related to the thing that have been requested by the user for the project. They are sometimes termed as functions or features of the system that will be provided by the project.

The organization of paper is alienated as section II describe about the definition of SDLC models, which are explained from the requirements point of view only, in section III a comparison based evaluation of SDLC models using 3D- bar graphs, section IV defines a comparison table and finally section V depicts the future work.

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

2.1 WATER FALL MODEL

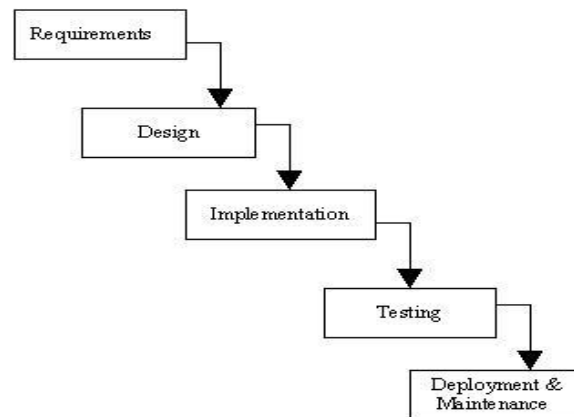


FIGURE 1. WATER FALL SDLC MODEL [1].

The requirements gathering process is intensified and focused specifically on software. To understand the nature of the program(s) to be built, the software engineer ("analyst") must understand the information domain for the software, as well as required function, behaviour, performance, and interface. Requirements for both the system and the software are documented and reviewed with the customer [2].

2. SOFTWARE PROCESS MODELS

The major weakness of the Waterfall Model as in figure 1. is that after project requirements are gathered in the first phase, there is no formal way to make changes to the project as requirements change or more information becomes available to the project team because requirements almost always change during long development cycles, often the product that is

• Manish Sharma, is currently pursuing PhD in Computer Science engineering in Graphic Era University, India, PH-+919634432297. E-mail: manish.sharma78@gmail.com

implemented at the end of the process is obsolete as it goes into production.[8] The 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 [8].

2.2 SPIRAL MODEL

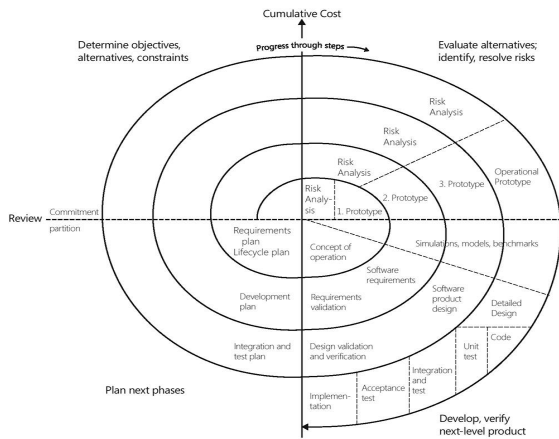


FIGURE 2. SPIRAL MODEL[2].

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 [8]. 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 [2][3].

2.3 RAD MODEL

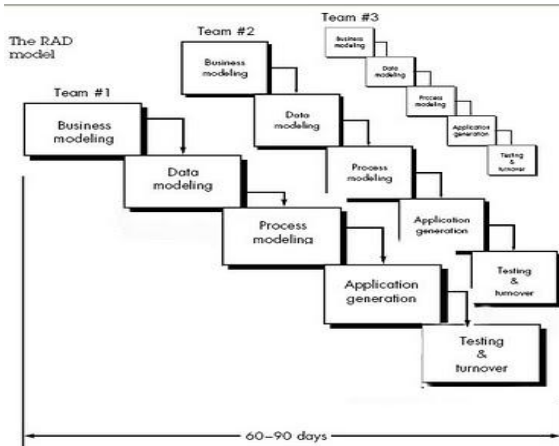


FIGURE 3. RAD MODEL[2].

If requirements are well understood and project scope is constrained, the Rapid application development (RAD) figure 3 process enables a development team to create a “fully functional system” within very short time periods (e.g., 60 to 90 days) [2].

2.4 INCREMENTAL MODEL

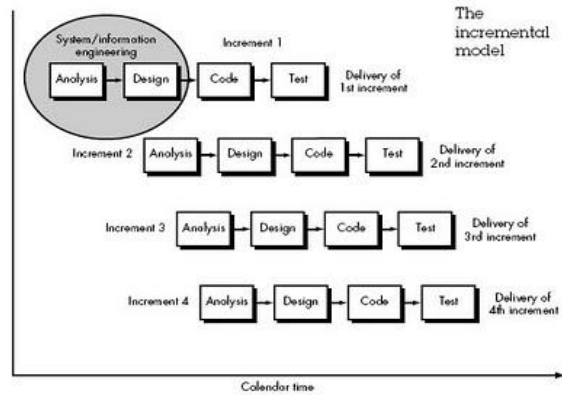


FIGURE 4. INCREMENTAL MODEL[2].

The incremental model figure 4 combines elements of the linear sequential model (applied repetitively) with the iterative philosophy of prototyping. The incremental model applies linear sequences in a staggered fashion as calendar time progresses. Each linear sequence produces a deliverable “increment” of the software when an incremental model is used; the first increment is often a core product [2].

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 [3].

3. SDLC COMPARISON TABLES

Project characteristic is measure in 0-10 rating. Comparison tables are design on three project characteristic categories.

1. Project Team
2. User Community
3. Project type and Risk

3.1 PROJECT TEAM

Whenever possible, it is best to select the people for the project team before selecting the 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:-

1. New to problem domain: - are the majority of team members new to the problem domain for the project?
2. New to the technology domain: - are the majority of the team members new to the technology domain for the project?
3. New to tools to be used: - are the majority of team member new to the tools to be used on the project?
4. Any training available: - is there training available for the project team, if required?
5. Comfortable with structure:- is the team more comfortable with structure than flexibility ?
6. Closely track by manager:- will the project manager closely track the team's progress ?

TABLE 1. COMPARISON BASED ON PROJECT TEAM

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	8

3.4 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:-

1. Availability of user representative restricted or limited: - will the availability of the user reprehensive be restricted or limited during the life cycle?
2. User representative new to the system definition:- are the user representatives new to the system definition?
3. User representative expert in problem domain:- are the user representatives expert in problem domain?
4. User representative want involve in SDLC:-do the user want to involve in all phases of the life cycle?

5. User representative want to track project progress: - does want customer want to track project progress?

TABLE 2. COMPARISON BASED ON USER COMMUNITY

User Community				
	WATER FALL	SPIRAL	RAD	INCREMENTAL
Availability of user representative restricted or limited.	9	2	1	7
new to the system definition	1	9	1	8
expert in problem domain.	1	2	9	8
want involve in SDLC.	1	1	9	2
want to track project progress.	1	9	2	1

3.5 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 simply provides a framework within which this action plan can be discussed and executed.

Characteristics of project type and risk:-

1. Integration project:- is the project a system integration project?
2. Enhancement to an existing system:- is the project an enhancement to an existing available project?
3. The funding for project:- is the funding for the project expected to be stable through-out the life cycle?
4. Project reliability:- Is the project high reliability a must?

TABLE 3. COMPARISON BASED ON PROJECT TYPE AND RISK

Project type and risk				
	WATER FALL	SPIRAL	RAD	INCREMENTAL
System integration project	1	8	7	9
Enhancement to an existing project	1	2	9	8
Funding is stable through out SDLC	9	2	8	1
High reliability must	1	9	2	8

4. CONCLUSION

What if during the course of the project something changes that cause the team to apply a different model that may be more appropriate? Can the model be changed during the execution of the project? The answer is, yes , it can be changed, but it should be done with careful consideration to the impacts of the project. Ultimately, it is better to change the model than to attempt to use one that is not well suited to meet the needs of the project.

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

1. Being familiar with the various models.
2. Review and analyze the types of work performed like development, enhancement, and maintenance.
3. Review the life cycle approach to standards required for your organization, your customer, or the type of project- ISO, IEEE, and so on.
4. Identify a set of phase and phase activates.
5. 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 AND RISK

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	Water fall
4.	Project reliability must	Spiral

5. ACKNOWLEDGEMENT

This research work is carried out with valuable support by Graphic Era Univesity, Dehradun, Uttarakhand, India.

6. REFERENCES

- [1] Ian Sommerville, “ Software Engineering”, 8th Edition, 2006, pp. 89.
- [2] R.S. Pressman, “*Software Engineering, A Practitioner’s Approach*”,5th ed. New York: McGraw-Hill, 2001, pp. 26.
- [3] B.W. Boehm, “*A Spiral Model for Software Development and Enhancement*”, IEEE, IEEE Computer Society, vol. 21, issue 5, May 1988, pp. 61 – 72.
- [4] 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.
- [5] Dinesh Kumar, Saroj Hiranwal” *Performance Enhancement of Software Process Models” 2010 2nd International Conference on Software Technology and Engineering(ICSTE)*.
- [6] Robert H. Martin, “*A Comparison of Software Process Modelling Techniques*”.