Critical Analysis of Software Process Models

Subodh Kumar
Department of Computer Applications,IET, Mangalayatan University, Beswan, Aligarh-U.P. (India)

N.K. Mishra
Department of Computer Applications, HIMCS, Farah, Mathura- U.P. (India)

Sarkar Sharan Mehta
Department of Computer Sc. & Engg., IET, DBRAU, Khandari Campus, Agra, U.P. (India)

ABSTRACT

Software process models are descriptive and diagrammatic form of the software life cycle. Software life cycle models provide a descriptive way to perform the various activities which are necessary during a software product development from scratch or even at some level of maintenance. In software process models the main focus has been drawn on the structured approach to build a new system or to improve an existing system. The basic activities of software development has been summarised in different life cycle models. The order of these activities in life cycle models are not homogeneous. However, there is not much variation between the software development structured process. On other hand, during software development there must be a sound understanding among project team members and they should have a clear-cut understanding about the various activities. Otherwise, unstructured process(without following any software process model) and lack of understanding between team would lead to project failure. Every life cycle model specify the entry and exit criteria in case of every development phase. Hence, developing a software product without software life cycle models not only difficult but also a pathetic approach which generally in-force towards project failure.

In this paper the performance various existing software process models such as Build And Fix, Waterfall Model, Rapid Application Development, Formal Systems Development Model, Prototyping Model, Incremental Model, Spiral Model, WIN WIN Spiral Model has been analysed on the basis of various features.

Keywords

Software Development Approach and SDLC Models

1. INTRODUCTION

Software Development Life Cycle (SDLC) is a process of building or maintaining software systems [4].software Process Model is an abstract representation of a software process [5]..All software projects go through the phases of requirements gathering, business analysis, system design, implementation, and quality assurance testing [6].There are several models for such processes, each describing approaches to a variety of tasks or activities that take place during the process. One life cycle model theoretically may suite particular conditions and at the same time other model may also looks fitting into the requirements but one should consider trade-off while deciding which model to choose[7].This research will provide a most efficient way to choose a better SDLC by comparing the most affecting features of the SDLC models.

2. PREVIOUS WORK

Various authors conducted a comparative study on analysis of different types of models[1,2,3,4] in software development life cycle and found that waterfall model is widely used by big companies for internal projects. A survey on software development life cycle models was done by [10, 12, and 14] and found that the waterfall model is the base for other software models. In present scenario many small and medium scale software companies are emerged and most of them companies are dealing with the web development related projects[15]. However, dealing with a software project is entirely related to the software development models as any software development may not be possible without using life cycle models.

3. ANALYSIS TABLE OF SOFTWARE PROCESS MODELS

Table 1.1: Analysis table of different life cycle models

Models → ↓ Features	Build &Fix	Waterfall	Formal System Development	Prototyping	RAD	Incremental	Spiral	WIN-WIN Spiral
Requirement analysis	Less Specified	Beginning of Project	Beginning	Changeable	Beginning	Intermediate	Reset At Each Spiral	Reset at Each Spiral
primary Objective Setting	Less Settled	Beginning of Project	More settled	Beginning of Project	Beginning of Module	At Beginning of Version	Before each Spiral	Before Each Spiral
Documentation	Worst	Well Documented	Worst	Not Good	Good	Well	Necessary	Necessary
Simplicity	Simple	Simplest	Complex	Complex	Simple	Less Complex	Complex	More Complex

Expensive	More	More	Medium	Less	Medium	Less	Medium	More
Appropriate Project Size	Small	Large	Small	Small	Medium	Large	Large	Large
Client Involvement	At Modifying	Beginning of Project	Very Less	Once in a Prototype	Beginning	Once in a Version	Much	Very Much

Table 1.1: Analysis table of different life cycle models (Continued)

Models → ↓ Features	Build &Fix	Waterfall	Formal System Development	Prototyping	RAD	Incremental	Spiral	WIN-WIN Spiral
Flexibility	Not	Rigid	Not	flexible	High Flexible	Less	High	Very High
Overlapping Phases	Yes	No	Yes	Yes	No	Not	Yes	Yes
Interleaved Phases	Yes	Sequential	Yes	Yes	No	Not	Yes	Yes
Applying changes	Costly	Very Costly	Costly	Less Costly	Not Costly	Costly	Costly	Less Costly
Phase Effect	Very Less	Very High	High	Low	Medium	High	High	Very High
Skelton Time	Quick	Very Long	Long	Quick	Quick	After First Version	After First Spiral	After First Spiral
Testing	after each Modification	In End of Project	At Last	In Each Iteration	In End of Module	At Each Version	Each Spiral	Each Spiral
Risk Abetment	Not	Only in Beginning	No	No	Yes	No	Yes, Each Spiral	Yes, Each spiral
Suitability for Risky Project	Not	No	No	Less	Medium	Less	Much	Very Much
Use of CASE Tools	Not	No	No	Yes	Yes	No	Yes	Yes
Running Condition	after each Modification	Last Phase	At Last	Every Prototype	Every Module	Each Version	Each Spiral	Each Spiral
Developer team	Small	Large	Small	Small Groups	Medium Groups	Large Groups	Large	Large
Cost Importance	Not	No	No	Yes	Yes	Not	Yes	Yes
time Importance	Not	No	No	Time Saving	Much Time Saving	Not	Yes	Yes
Quality Level	Worst	High	Average	Medium	Medium	High	Medium	High

Table 1.1: Analysis table of different life cycle models (Continued)

Models → ↓ Features	Build &Fix	Waterfall	Formal System Development	Prototyping	RAD	Incremental	Spiral	WIN-WIN Spiral
Targeted Project Type	Simple	Gov. Qualitative	Applied	mostly Embedded	Business Applied	Large Projects	Technical Risky Projects	More Risky Projects
Implementation	After Full Modified	After Completion	At Last	After Completion	At Each Module	At Each Version	After All Spirals	After All Spirals
Successful Project	Worst	Less	Less	Medium	Medium	Medium	High	Very High

Rate								
Developer Experience	High	Medium	High	High	Medium	High	High	High
Client Feedback	At Each modificati on	No	No	After Prototyp e	At Last of Module	At Each Iteratio n	In Each Spiral	In Each Spiral

4. CONCLUSION

From the analysis table it is clear that the waterfall model is widely acceptable and used in government, industrial projects as in this model experienced manpower deals project with the full requirement specification. The Spiral Model more beneficial when it processes large and technically complex projects whose development needs extensive risk abatement. The WIN-WIN Spiral model is also good for complex and risky projects. This model also allows much involvement of customer than in simple spiral model. The prototype approach is much effective when customers are not clear with their requirements. The rest of other models considered in this study are not playing vital role in software development hence they are in less use.

5. REFERRENCES

[1] Nabil Mohammed Ali Munassar & A. Govardhan, *a comparison between five models of software engineering*, IJCSI Issues, Vol. 7, Issue 5, 2010

[2] Shikhamaheshwari, Dinesh Ch. Jain,*A Comparative Analysis of Different types of Models in Software Development Life Cycle*, IJARCSMS, Volume 2, Issue 5, 2012

[3] T Bhuvaneswari, S Prabaharan, *a Survey on Software Development Life Cycle Models*, IJCSMC, Vol. 2, Issue. 5, 2013, pg. 262 – 267.

[4] *SystemsDevelopment Lifecycle: Objectives and Requirements*. Bender RPT Inc, 2003.

[5] Ian Sommerville, "*Software Engineering*", 8th Edition, 2006, pp. 89.

[6] Klopper, R., Gruner, S., &Kourie, D. "*Assessment of a framework to compare software development methodologies*", 2007

[7] *Software Methodologies Advantages & disadvantages of various SDLC models.mht*, www.ijarcsse.com

[8] Nabil Mohammed Ali Munassar1 and A. Govardhan, *A Comparison Between Five Models Of Software Engineering*, IJCSI, *Vol. 7, Issue 5, September 2010*

[9] Roger Pressman, *Software Engineering: A Practitioner's Approach*, Sixth Edition, McGraw-Hill Publication

[10] Vishwas Massey, K.J Satao, *"Comparing Various SDLC Models And The New Proposed Model On The Basis Of Available Methodology".*

[11] Barry Boehm, *"Spiral Development: Experience, Principles, and Refinements"*, edited by Wilfred J.Hansen,2000

[12] Craig Larman and Victor Basili, *Iterative and Incremental Development: A Brief History*, IEEE Computer, June 2003.

[13] Apoorva Mishra, DeeptyDubey, *A Comparative Study of Different Software Development Life Cycle Models in Different Scenarios*, IJARCSMS, Volume 1, Issue 5, 2013.

[14]. T. Bhuvaneswari, S. Prabaharan. *A Survey on Software Development Life Cycle Models*, IJCSMC, 2013.

[15.] P. Allen, M. Ramachandran, H. Abushama, PRISMS: an approach to software process improvement for small to medium enterprises, Proceedings of Third International Conference on Quality Software, 2003.