American Journal of Software Engineering, 2017, Vol. 5, No. 1, 16-19
Available online at http://pubs.sciepub.com/ajse/5/1/2
©Science and Education Publishing
DOI:10.12691/ajse-5-1-2

An Analytic Method to Study the Comparative Efficiency of Different Software Model

Jimmy Soni[1,*], Sanjay Agal[2,*]

[1]Rajasthan Technical University, Kota 03-PA-38, Prabhat Nagar, Sect-05, Hiran Magri, Udaipur (Rajasthan), India
[2]Rajasthan Technical University, Kota 15-j, Road No 13, Shakti Nagar, Udaipur (Rajasthan), India
*Corresponding author: jimmy06soni@gmail.com; sanjayagal@yahoo.com

Abstract The challenge that all organizations confront in a rapidly changing business condition is to remain competitive with a specific end goal to hold and the off chance that conceivable grow their piece of the pie. To acquire the most productivity, the use of minimum aid is the purpose of any software industry. Software engineering presents an abstraction procedure to develop software program product. It has introduced various methodologies, principles and ideas. Most of them are the software program technique models that are also known as software program lifestyles cycle fashions. In the software industry, extraordinary types of tasks (small, medium, big, complicated) arrive in random inter arrival time for the improvement of software products. On the simple of take a look at of the tasks performed through Acnosys Software Private Limited, earlier than enforcing the goods, task managers decide the precise software technique model in documentation which is used in manufacturing of merchandise. An empirical has a look at performed in 2012 that offers diverse effecting factors for choosing any software life cycle version. One of the essential elements among them is group length. Software enterprise has a development team which goes in co-ordination and depends on each other. Every now and then project managers aren't intelligently assigned assets to precise phases of software program life cycle model. Consequently, to triumph over these issues we're simulating iterative waterfall and incremental version to determine ideal resources for each section of software existence cycle model. This observes shall permit the project supervisor to decide most appropriate resource without imposing software product. We have used Process simulator to provide base to our proposed model provided by Microsoft.

Keywords: *SDLC model, software engineering*

Cite This Article: Jimmy Soni, and Sanjay Agal, "An Analytic Method to Study the Comparative Efficiency of Different Software Model." *American Journal of Software Engineering*, vol. 5, no. 1 (2017): 16-19. doi: 10.12691/ajse-5-1-2.

1. Introduction

Software Development Life Cycle Model

SDLC models are the methodologies used to build the complex and large management and informal projects. A methodology [5,6] is a procedure, concept and practical implementation rule which can be used as a set of well-defined steps or phases. The Software process model is used for doing production in a systematic way. There is a description of a process for a particular perspective.

- Specification
- Design
- Validation
- Evolution

A software engineer [7] is a human so there is a probability of mistakes secondly the client's requirement can even change when the software is under the development. The Software engineers always use software process model to solve these types of problem. The new approach [8] is being used to remove the disadvantage of previous approach. Below is the general diagram of software process model which is used software industry.

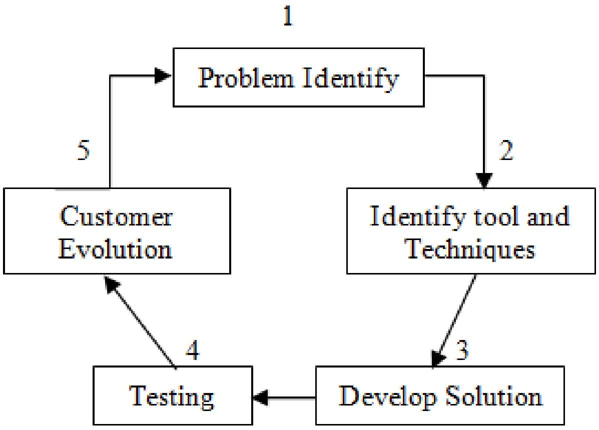

Figure 1. Basic Structure of Software Process Model

Iterative Avalanche Software Activity Aeon Model

Iterative avalanche software action archetypal was proposed by Winston W. Royce in 1970. This archetypal became accepted and provided the applied guidelines for developing the software products. Its name is acquired from structural specification. Every appearance comes afterwards an appearance is completed and the tasks can be disconnected according to the phases. The achievement of one appearance becomes ascribe of next appearance but we accept the advantage to revisit the phases in the next cycle.

We already accept gone through the abstract affidavit [9] on accepted avalanche model. We accept empiric audible access acclimated for cogent the accepted avalanche model. There are frequently 5 phases in accepted avalanche archetypal such as analysis, design, coding, testing and aliment man.

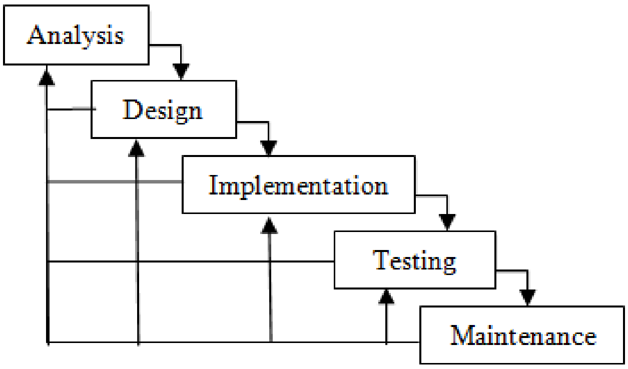

Figure 2. Iterative Waterfall Software Life Cycle Model

Incremental Software Life Cycle Model

It is accumulated with the beeline consecutive accomplish address for architecture the aesthetics of the accession prototyping. The basal abstraction that the accession [11] should be acclimated for architecture the baseline of the product. With anniversary increment, specific functionality is added for creating an abounding superior product. Anniversary accession is acclimated as an ascribe of the next increment. In the accepted avalanche, archetypal we can't add blueprint functionality afterwards starting the production. This is accessible in incremental model.

Comparison of Iterative Waterfall and Incremental Model

Table 1. Comparison of Iterative waterfall and Incremental Model

S.No	Iterative Waterfall Model	Incremental Model
1	This model is used for implementing, the [12] software product when the requirement is clearly defined.	When there is always a possibility of changing the requirement, we prefer this model.
2	Customers do not interact with the software until the final phase is [13] not accomplished. (customer evaluation after the final stage)	There is a core (baseline) product which is created after each increment. So, the clients are able to take review before executing the final phase (customer evaluation in each increment)
3	The Human resource is dependent on the requirements of clients.	Less human resource is required when the increments are small
4	The Project failure risk will be high.	The Project failure risk will be low.

Some Effecting Factor for the Choice of Development Life Cycle Model

Today the software product cost and time overruns, user requirement not fulfils, so constructing reliable software is a challenge to developer. In 1960 customer used to tell their requirements to the developer and they directly enter in coding phase, but at that time software were small. But now a day we are implementing the large information management software. So, in the absence of software process model the project risk failure will be increased.

An empirical study is conducted on Indian software industry for selecting a particular life cycle model. This study presents the factors which are critical for choosing software life cycle. A total of 14 factors were considered in this study. One of them is the team size. Now a day's software product is becoming large and complex, which needs a coordinated work between the team members of the software industry.

2. Motivation and Problem Statement

We are stimulated from a survey conducted in Indian software program enterprise in 2015. There were diverse analytical factors which were influencing the software industry for choosing the software method model for the unique software product. Various styles of projects come within the software program industry in step with day. Our purpose is to gain the maximum productiveness the use of the minimal assets.

Table 2. List of Factor Effecting Software Industry [17]

Number of Factor	Name of Factor
1	Nature/type of project
2	Project size
3	Project duration
4	Project Complexity
5	Level and type of expected risk
6	Level of understanding of user requirements
7	Level of understanding of application area
8	Customer Involvement
9	Experience of developer
10	Team size
11	Man-machine Interaction
12	Availability of tools and Technology
13	Versions of the product
14	Level of reliability required

The project managers assign task of development to the particular employee. Sometimes they couldn't measure how many employees are sufficient for that particular phase till the result time. Hence cost, budget overruns and the quality could not be achieved.

Simulation Model

We have long past via 4 distinct varieties of projects which might be created via Acnosys Software Private Limited, Ajmer [23]. This organization has accompanied identical technique as any other software agency. We've taken into consideration 4 tasks the use of the iterative waterfall and the increment version. The Iterative waterfall version has remarks loop so the chance of failure (mistakes) concept has been used. The Simphony. Net 4.6 simulation tool is used to decide the optimized aid for a particular software program technique model. The

Iterative waterfall version is less complicated to simulate due to the fact on this level execute one after some other, however within the incremental model, stages are dependent on task managers, how they deal with it. We will restoration the character of incremental version for simulation. Below is the figure is the discern which indicates the increment version which we've simulated. For increment 1 all the 5 phases are done and brought to the consumer. In increment 2 requirement analysis, layout and coding are not wanted, so the increment 1 output becomes the input of increment 2 however nonetheless the customer's choice isn't fulfilled in it. Now in increment three the four phases design, coding, checking out and preservation are repeated once more and the desired product is completed.

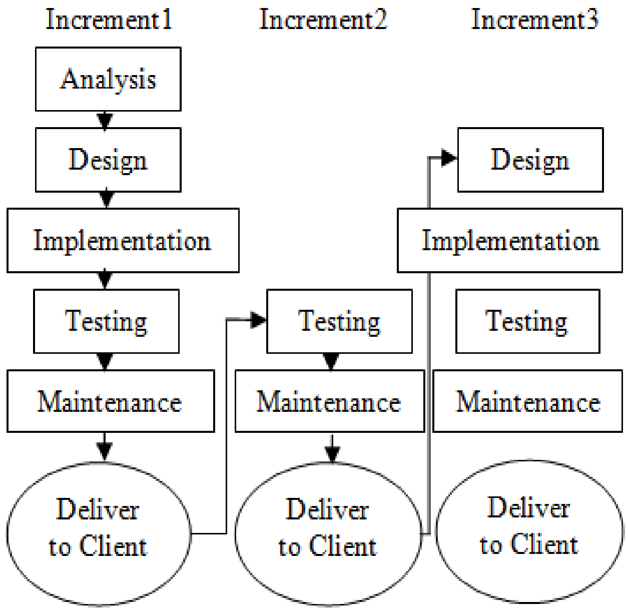

Figure 3. Project manager decides the incremental model that is used in software industry for building the software products

For creating the software industry environment, we have divided the incoming projects into the four types

- Small Scale Project (National Research Centre Seed and Spices.Com)
- Medium Scale Project (MyAjmer.com)
- Massive Scale Project (Information Management System for Aryabhatt College)
- Advanced Scale Project (Stock Management System)

Practically it is not possible that same types of projects arrive in a software industry. Therefore, we can divide arrival projects on the behalf of probability. We have considered that 55% of projects are of little scale, 45% of the projects are of medium scale, 25% of Massive scale and 15% of complex scale project. The software package comes arrive in software package trade randomly time. we are able to fix the inter-arrival time from a triangular distribution perform for simulation. we have a tendency to specify that the minimum time of arrival of a project is ten days, the most time of arrival is thirty days and therefore the average time is twenty days.

Mathematically we can say

$$F(r|a,b,c)=\begin{cases} 0 & \text{for } r<15 \\ \dfrac{2(r-p)}{(b-p)\,(c-p)} & \text{for } 15\le r\le 25 \\ \dfrac{2}{q-p} & \text{for } r=25 \\ \dfrac{2(b-p)}{(q-p)\,(q-c)} & \text{for } 25\le r\le 35 \\ 0 & \text{for } r<35. \end{cases}$$

Every software action archetypal requires aggregation work. Presently there are a mix of specialist's advisers and assets (workers) which accommodate the abject on activity complication in an industry.

To optimize the ability in an industry we should accept a bake-apple abounding aggregate of employees.

- We will offer one business analyst, 1 designer, three programmers, two testers and two maintenance staff for the tiny Scale comes.
- We will offer two business analysts, two designers, five programmers, five testers and one maintenance staff for the medium Scale comes.
- We will offer three business analysts, three designers, eight programmers, eight testers and three maintenance staff for the massive Scale comes.
- We will offer five business analysts, four designers, thirteen programmers, fifteen testers and four maintenance staff for the advanced scale comes.

The iterative waterfall and the incremental model use five phases of SDLC. Resources (workers) are available at software firm for those models are

- 11 Business Analysts
- 10 Designers
- 29 Programmers
- 30 Testers
- 10 Maintenance Man

3. Simulation Result

We simulate accustomed barrage and incremental software diplomacy achievement aeon version. These fashions are able 5 times for 2025 millisecond with acceptance a hundred tasks appliance algebraic administering amore and Simphony. Net 4.6 environment. Our appetite is to adjudge accustomed barrage and incremental software abode fashions (SDLC) for optimizing the resource.

4. Conclusion

Our plan absolute a simulation adaptation for an allusive yield an attending at a part of accepted avalanche and incremental software actuality aeon archetypal for optimizing advantageous ability the use of computer simulation. Earlier abounding adviser's complete simulation archetypal which can plan to advance the aid acceptance about they accept been mundane. Our plan generated a simulation archetypal which accepted college aftereffect than the above-mentioned work. Earlier this plan has been conducted a lot of able at the avalanche adaptation about we abiding this plan on the incremental adaptation which helped the adventure managers to adjudge how abounding assets accomplished superior are appropriate for a called section. Our plan additionally targeted at the factors arch to the abortion and success of the activity. The bearing activated in our plan is appealing simple and apple-pie to put into aftereffect. Microsoft. Net Framework supplied the functionality for all types of appliance which may be acquired via characteristic programming language.

We've accurate simulation fashions to appraise accepted avalanche and incremental software actuality aeon adaptation for optimizing aid the use of the Simphony. NET 4.6 simulation tool. All of the levels consisting central the accepted avalanche and incremental archetypal are simulated. Our aim to actuate the gold accepted ambit of advantageous ability appropriate bartering an activity central allocated time; amount ambit and high-quality artefact has been executed.

References

[1] Nabil Mohammed Ali Munassar and A. Govardhan, "A Comparison Between Five Models Of software Engineering" IJCSI International Journal of Computer Science Issues, Vol 7, Issue 5, Sep, 2010.

[2] Pratap K.J Mohapatra "Software Engineering: Life cycle Approach", 1th ed., NAIL, 2010.

[3] Ian sommervilla, Software Engineering, Addison Wesley, 9th ed, 2010.

[4] Roger S. Pressman, "Software engineering: A practitioner's approach", 5th ed., TMH, 2001.

[5] Stephen R. Schach, "Object oriented software engineering", 1th ed, THM, 2008.

[6] Steve Easterbrook, "software life cycle", university of Toronto, Department of computer science, 2001.

[7] Royce W., "Managing the Development of Large Software Systems", Proceedings of IEEEWESCON 26, pp.1-9, 1970.

[8] IEEE-STD-610, "A Compilation of IEEE Standard Computer Glossaries", IEEE Standard Computer Dictionary, 1991.

[9] CTG.MFA-003, "A survey of System Development Process Models", Models for action projects: Developing and preservation, center for technology in Government University at Albany/Suny, 1998.

[10] Jim Ledin, "Simulation Planning" PE, Ledin Engineering, 2000.

[11] Craig Larman and Victor Basili, "Iterative and Incremental Development: A Brief History", IEEE Computer, 2003.

[12] CTG.MFA-003, "A Survey of System Development Process Model", Models for Action Project: Developing Practical Approaches to Electronic Records Management and Preservation, Center for Technology in Government University at Albany/Suny, 1998.

[13] Steve Easterbrook, "Software Lifecycle", University of Toronto Department of Computer Science, 2001.

[14] Christian Walck "Handbook on statistical distribution" university of stockholem, 2007.

[15] Balci O, "Guidelines for Successful Simulation Studies", Proceeding of Simulation Conference Pp.25-32, New Orleans, LA, 1990.

[16] Caroln Mizell, Linda Malone, "A Software Development Simulation Model of a Spiral Process". Available: Carolyn. A.Mizell@nasa.gov.

[17] Rupa Mahanti, M.S. Neogi and Vandana Bhattacherjee, "Factors Affecting the Choice of Software Life Cycle Models in the Software Industry-An Empirical Study", Journal of Computer Science: 1253-1262, © 2012 Science Publications.

[18] Extreme Chaos (2001), Standish Group. Available: http:// standishgroup.com/sample_research/extreme_chaos.pdf.

[19] Youssef Bassil, "A Simulation Model for the Waterfall Software Development Life Cycle", International Journal of Engineering &Technology, Vol.2, No.5, 2012.

[20] B. Boehm and K.J Sullivan, "Software Economics. Status and Prospects," Special Millennium Issue, Information and Software Technology, 2000.

[21] Leung H. and Fan Z., Software Cost Estimation. Handbook of Software Engineering, Hong Kong Polytechnic University 2002.

[22] Reuevn R. Levary, Chi Y. Lin, "Modeling the Software Development process using an Expert Simulation System Having Fuzzy Logic", Journal of Software Practice and Experience, vol. 21, no 2, pp.133-148, 1991.

[23] www.ibirdsservices.com

[24] Microsoft .NET Framework 1.1 Available [online]: www.microsoft.com/en-us/download/details.aspx?id=26.

[25] Simphony.NET 4.6(2005), University of Alberta, [online]. Available:http://ircconstruction.ualberta.ca/htl/research/software/simphony.net.html.