A Comparative Study between Iterative Waterfall and Incremental Software Development Life Cycle Model for Optimizing the Resources Using Computer Simulation

Prakriti Trivedi[#1], Ashwani Sharma[*2]

Assistant Professer CS and IT
Govt.Engineering College
Ajmer, Rajasthan India
[1]niyuvidu@rediffmail.com
M.Tech (IT), Govt.Engineering College
Ajmer, Rajasthan India
[2]ashwnaipareek90@gmail.com

***Abstract*- To achieve the maximum productivity using minimum resource is the aim of any software industry. Software engineering provides an abstraction process to develop software product. It has introduced various methodologies, principles and concepts. Most of them are the software process models which are also known as software life cycle models. In the software industry different-different types of projects (small, medium, large, complex) arrive in random inter arrival time for development of software products. Before implementing the products, project managers decide an appropriated software process model in documentation that is used in production of products. An empirical study conducted in 2012 that presents various effecting factors for selecting any software life cycle model. One of the important factors is team size. Software industry has a development team which works in co-ordination and depends on each other. Sometimes project managers are not intelligently assigned resources to particular phases of software life cycle model. Therefore to overcome these issues we are simulating iterative waterfall and incremental model to determine optimal resources for every phase of software life cycle model. This study shall enable the project manager to determine optimal resource without implementing software product. As future work, we shall be simulating spiral and iterative incremental model to determine which SDLC model is more appropriate for software industry**.

***Keywords*- Science; Software engineering; Software process model; Iterative waterfall model; Increment model; Simphony model; Simphony.NET**

I. INTRODUCTION

We can divide science into two aspects; one is pure science and second is applied science. Pure science involves theoretical concepts and principles. This practical implementation is known as Engineering. Computer Science falls in the category of pure science. We usually study only the theoretical literatures in it. Software engineering is one of applied fields of computer science [1].It has methodology and principles which can be applied in real scenario of software Industry. In ancient times people didn't have ways to express their feelings. They needed framework which could help. 1960's software engineering [2] term coined clients didn't follow any particular process to develop software product. The clients used to tell their requirement [3, 4] manually to the project manager and they directly fixed code. As time spent the requirement of the clients [5] also got increased. Now we are conducted work with large, complex information and management software systems, examples are banking, healthcare, telecommunication and defence etc. We need an abstraction process for building a software product. So, the Software engineering is related with the concept, process and tools which are required to make software. The goal of software engineering is to produce the software within the estimated budget, delivered [6] on time and caters the user's requirement.

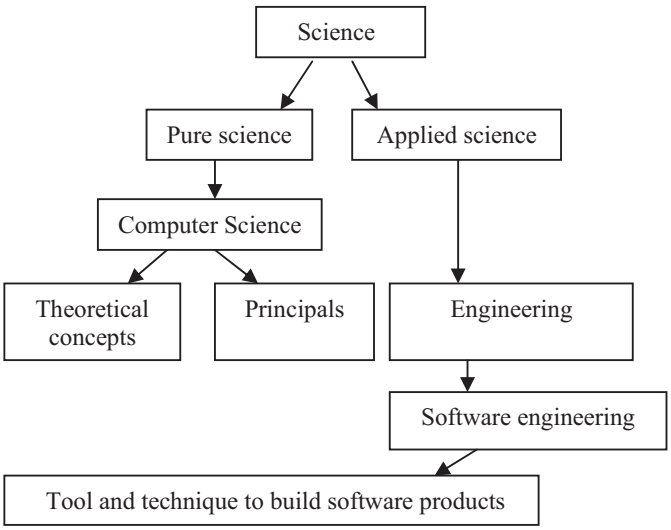

Fig 1 Overview software engineering

Software process model (SDLC) is a framework [7,8] it is a pre-planning of any software industry that tells how to develop a software product, there are some effective factors these are involved for selecting any software process model. We shall be studying team size factor with Microsoft 1.1 framework environment and Simphony.NET 1.1.3.14 simulation tool. We shall be simulating iterative waterfall and incremental model under some assumption.

II. ITERATIVE WATERFALL SOFTWARE LIFE CYCLE MODEL

Iterative waterfall software process model was proposed by Winston W.Royce in 1970. This model became popular and provided practical guidelines for developing software product. Its name is derived from structural specification. Every phase comes after a phase is completed and tasks can be divided according to phases. The output of one phase becomes input of next phase but we have the option to revisit phases in the next cycle.

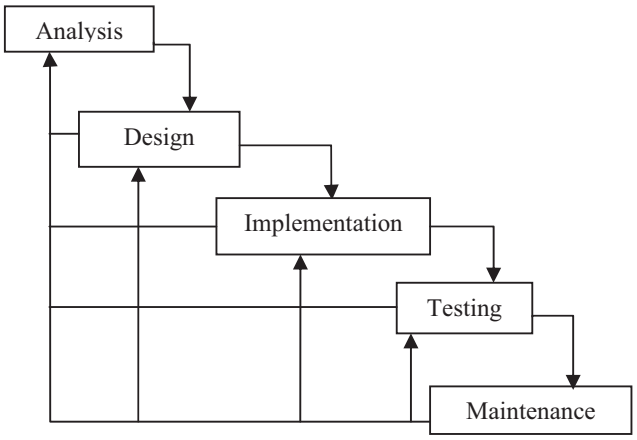

Fig 2 Iterative waterfall software process model

III. INCREMENTAL SOFTWARE LIFE CYCLE MODEL

It is combined with the linear sequential steps technique for building the philosophy of the increment prototyping. The basic idea that increment [9] should be used for building the baseline of the product. With each increment specific functionality is added for creating a full quality product. Each increment is used as an input of the next increment. It is used where the customer requirement is frequently changing. Initially the developer and the customer communication takes place, customer tells about requirement and technology which is required. The developer understands requirements of the client and starts building the product. Now the first increment is completed and the product is delivered to the customer and put into the operational environment for testing the conformance of requirements.

IV. TECHNOLOGY

A. Microsoft .Net Framework

Microsoft announced a new software [10] development framework for window operating system in July 2000.It has put effort on .NET framework because future application depends on its success, in other words it represents an evolution as DOS to Window. In the future, new and upgraded versions of existing application would be depending on .NET framework.

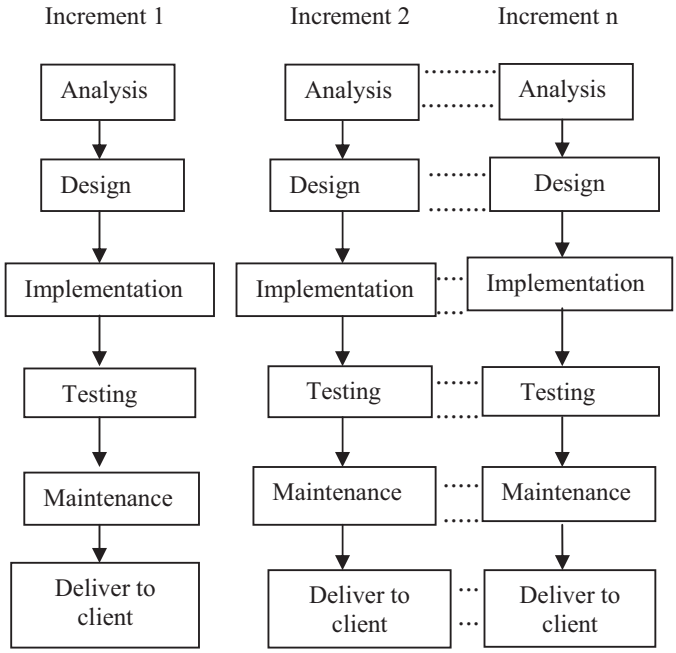

Fig 3 Incremental software process model

B. Simphony .NET 1.1.3.14

It is next evolution of computer simulation in simulation area. There is a large library of modelling elements that are available with the basic distribution function .Simphony.NET [11] is a simulation tool of software industry.

V. MOTIVATION AND PROBLEM STATEMENT

We are inspired a survey [12] conducted in Indian software industry in 2012. There were various analytical factors which are influencing software industry for selecting the software process model for the particular software product. Various types of projects come in the software industry per day. Our goal is to achieve maximum productivity using minimum resources.

The project managers [13] 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 cannot be achieved.

Table 1 LIST OF FACTORS EFFECTING SOFTWARE INDUSTRY

Number of factor	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 the application area
8	Customer involvement
9	Experience of developers
10	Team size
11	Man-machine interaction
12	Availability of tools and technology
13	Versions of the product
14	Level of reliability required

A Effect on Software Industry

1) The project managers have no idea that how many resources are sufficient for a particular phase. Due to the improper resource and unintelligent assigning of the employees SDLC phase [14] may be delayed due to other dependent phases may stay idle.

2) Reasons for project failure

- Project get over budget
- Project is unable to fulfill needs of customers

VI. ASSUMPTION AND SPECIFICATION FOR SIMULATON MODEL

Below are the assumptions and specifications for both the iterative waterfall and the increment model. The iterative waterfall model have feedback loop so the probability of failure (error) concept can be used. The Simphony.NET simulation tool is used to determine the optimized resource for a particular software process model. The iterative waterfall model is easier to simulate because in it phases [15, 16] execute one after another, but in the incremental model phases are dependent on project managers, how they deal with it. We can fix the nature of incremental model for simulation. Below is the figure which shows the increment model which we have simulated. Let us assume in increment 1 all the five phases are executed and delivered to the client. In increment 2 requirement analysis, design and coding are not needed, so increment 1 output becomes the input of increment 2 but still the customer desire is not fulfilled in it. Now in increment 3 the four phases design, coding, testing, maintenance are repeated again and the desired product is achieved

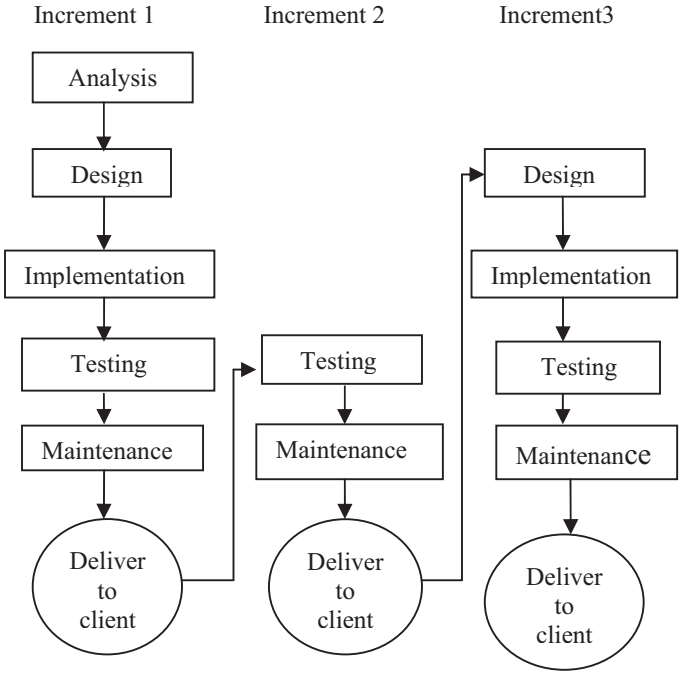

Fig 4 Project manager decide incremental model that use software industry for build software product

For creating software industry environment we divided the incoming project into four types-

- Small Scale Project
- Medium Scale project
- Large Scale project
- Complex Scale project

This is not practical that only the same type of projects come in a software industry. Therefore we can divide arrival projects on the behalf of probability. Let as assume that 45% of projects are of small scale, 35% of the projects are of medium scale, 15% of large scale and 5% of complex scale project. The Software projects arrive in software industry at random time. We can fix the inter-arrival time from a triangular distribution function for simulation .We specify that the minimum arrival time of a project is 10 days , the maximum arrival time is 30 days and the average time is 20 days.

Mathematically we can say

$$f(x|a,b,c) = \begin{cases} 0 & for\ x < 10 \\ \frac{2(x-a)}{(b-a)(c-a)} & for\ 10 \le x \le 20 \\ \frac{2}{b-a} & for\ x = 20 \\ \frac{2(b-a)}{(b-a)(b-c)} & for\ 20 \le x \le 30 \\ 0 & for < x \end{cases}$$

Every software process model requires team work. Presently there are mix of specialist's employees and resources (workers) which provide the base on project complexity [17] in an industry.

To optimize the resource in an industry we should have a fruit full combination of employees.

- We can provide 1 business analyst, 1 designer, 2 programmers, 2 testers and 1 maintenance employees for Small Scale projects.
- We can provide 2 business analysts, 2 designers, 4 programmers, 5 testers and 1 maintenance employees for Medium Scale projects.
- We can provide 3 business analysts, 3 designers, 7 programmers, 8 testers and 3 maintenance employees for Large Scale projects.
- We can provide 4 business analysts, 4 designers, 12 programmers, 15 testers and 4 maintenance employees for complex scale projects.

The iterative waterfall and the incremental model use five phases of SDLC. Let as assume that resources (workers) available at software firm for those models are:-

- 10 Business Analysts
- 10 Designers
- 25 programmers
- 30 Testers
- 9 Maintenance Man

Software process models have a team of specialists which build small, medium, large and complex projects. This is defined below by uniform distribution [12, 13] for creating simulation models. We also define uniform distribution function with formula for only small scale projects.

A. Small scale project assumed time

The business analysis phase (small scale) requires a uniform distribution with a lower limit of 2 days and an upper limit of 3 days.

$$F(x) - \begin{cases} \frac{1}{b-a} & for\ 2 \le x \le 3 \\ 0 & for\ x < 2\ or\ x > 3 \end{cases}$$

The Design phase (small scale) requires a uniform distribution with a lower limit of 3 days and an upper limit of 5 days.

$$F(x) = \begin{cases} \frac{1}{b-a} & for\ 3 \le x \le 5 \\ 0 & for\ x < 3\ or\ x > 5 \end{cases}$$

The Implementation phase (small scale) needs a uniform distribution with a minimum time of 5 days and a maximum of 7 days.

$$F(x) = \begin{cases} \frac{1}{b-a} & for\ 5 \le x \le 7 \\ 0 & for\ x < 5\ or\ x > 7 \end{cases}$$

The Testing phase (small scale) requires a uniform distribution with a lower limit of 2 days and an upper limit of 6 days.

$$F(x) = \begin{cases} \frac{1}{b-a} & for\ 2 < x < 6 \\ 0 & for\ x < 2\ or\ x > 6 \end{cases}$$

The maintenance phase (small scale) requires a uniform distribution with a lower limit of 1 days and an upper limit of 2 days.

$$F(x) = \begin{cases} \frac{1}{b-a} & for\ 1 \le x \le 2 \\ 0 & for\ x < 1\ or\ x > 2 \end{cases}$$

B. Medium scale project assumed time

The business analysis phase (medium scale) requires a uniform distribution with a lower limit of 4 days and an upper limit of 6 days.

The Design phase (medium scale) requires a uniform distribution with a lower limit of 4 days and an upper limit of 7 days.

The Implementation phase (medium scale) needs a uniform distribution with a minimum of 9 days and a maximum of 18 days.

The testing phase (medium scale) requires a uniform distribution with a lower limit of 3 days and an upper limit of 7 days.

The maintenance phase (medium scale) requires a uniform distribution with a lower limit of 3 days and an upper limit of 6 days.

C. Large scale project assumed time

The business analysis phase (large scale) requires a uniform distribution with a lower limit of 5 days and an upper limit of 9 days.

The Design phase (large scale) requires a uniform distribution with a lower limit of 10 days and an upper limit of 20 days.

The Implementation phase (large scale) needs a uniform distribution with a minimum of 25 days and a maximum of 30 days.

The Testing phase (large scale) requires a uniform distribution with a lower limit of 10 days and an upper limit of 15 days.

The Maintenance phase (large scale) requires a uniform distribution with a lower limit of 6 days and an upper limit of 9 days.

D. Complex scale project assumed time

The Business analysis phase (complex scale) requires a uniform distribution with a lower limit of 10 days and an upper limit of 17 days.

The Design phase (complex scale) requires a uniform distribution with a lower limit of 20 days and an upper limit of 30 days.

The Implementation phase (complex scale) needs a uniform distribution with a minimum of 35 days and a maximum of 45 days.

The Testing phase (complex scale) requires a uniform distribution with a lower limit of 15 days and an upper limit of 25 days.

The Maintenance phase (complex scale) requires a uniform distribution with a lower limit of 10 days and an upper limit of 16 days.

E .Iterative waterfall model probability failure

We have assumed that 10% probability in the small scale project of an error.
We have assumed that 20% probability in the medium scale project of an error.
We have assumed that 30% probability in the large scale project of an error.
We have assumed that 40% probability in the complex scale project of an error.

VII SIMULATION RESULTS

We simulate iterative waterfall and increment software life cycle model. These models are executed 5 times for 1825 millisecond with incoming 100 projects using mathematical distribution function and Simphony.NET environment. Our goal is to compare iterative waterfall and incremental software process models (SDLC) for optimizing resource.

We denote graphical representation (time graph) between required optimized resource (Entities) and Simulation time

A .Optimal resources analysis for iterative waterfall model

1).*Time graph for Analysis phase:* In figure 5,this graph is generated for analysis phase in iterative waterfall model.The required optimal resource for businees analysis is 8.60 which make a continous flow of a software project in the industry. This numeric value show the minimum required resource for maximum productivity.

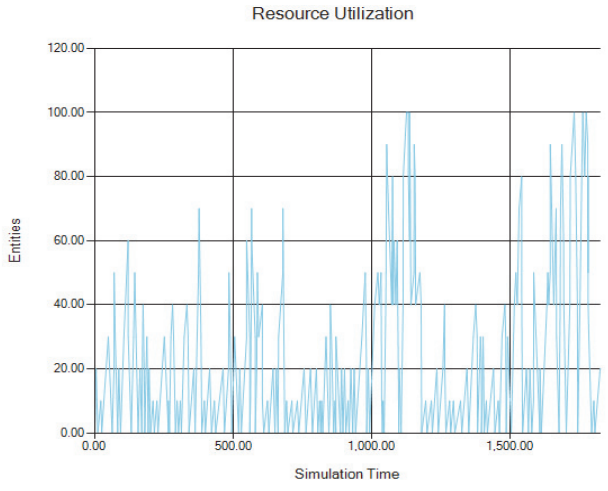

Fig 5Utilization of business Analyst

*2).Time graph for Design phase***:** In figure 6,this graph is generated for Design phase in iterative waterfall model.The required optimal resource for desginer is 20.10 which make a continous flow of a software project in the industry.This numeric value show the minimum required resource for maximum productivity

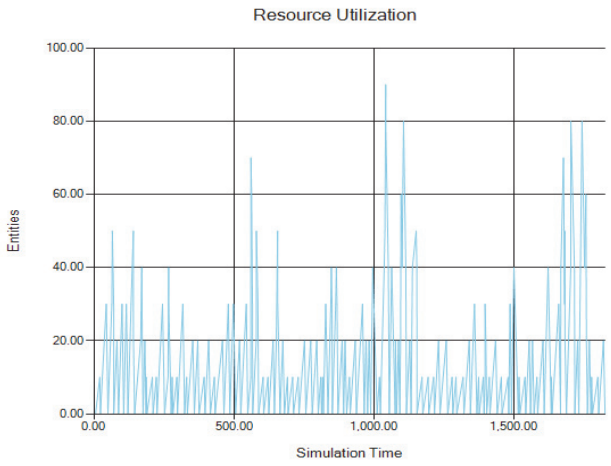

Fig 6 Utilization of Designer

3)Time graph for Implementation phase: In figure 7,this graph is generated for implementation phase in iterativewaterfall model.The required optimal resource for programmer is 33.81 which make a continous flow of a software project in the industry.This numeric value show the minimum required resource for maximum productivity

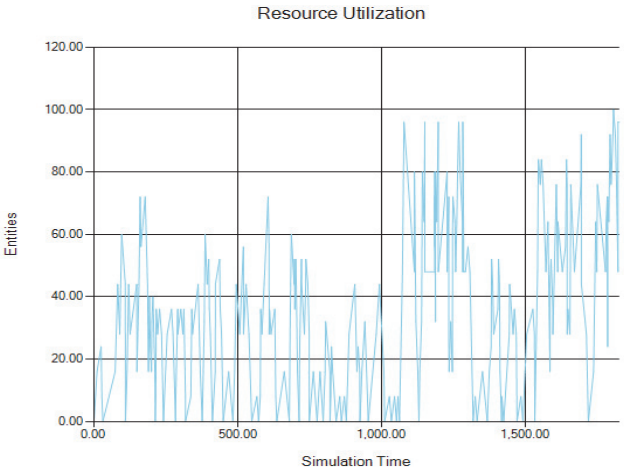

Fig 7Utilization of Programmer

4) Time graph for testing phase: In figure 8,this graph is generated for testing phase in iterative waterfall model.The required optimal resource for tester is 13.32 which make a continous flow of a software project in the industry.This numeric value show the minimum required resource for maximum productivity.

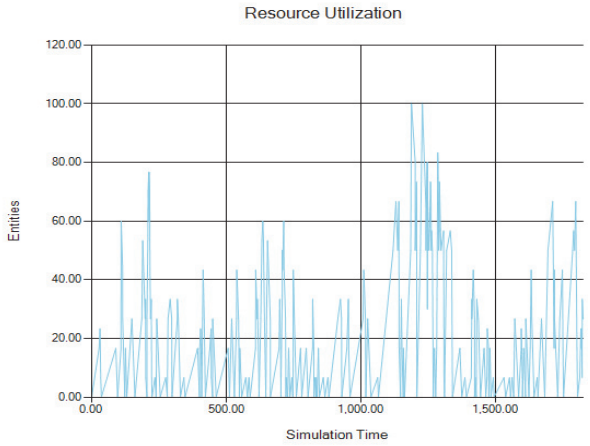

Fig8 Utilization of tester

5) *Time graph for maintenance phase*: In figure 9,this graph is generated for maintenance phase in iterative waterfall model.The required optimal resource for maintenance man is 5.78 which make a continous flow of a software project in the industry.This numeric value show the minimum required resource for maximum productivity.

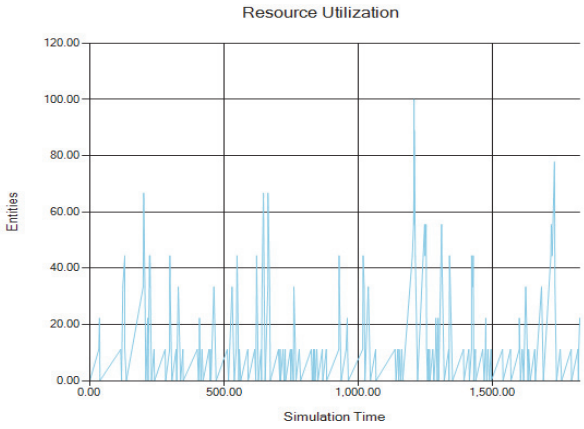

Fig 9 Utilization of Maintenance Man

B Optimal resources analysis for incremental model

1) Time graph for Analysis phase: In figure 10,this graph is generated for analysis phase in incremental model.The required optimal resource for businees analysis is 2.27 which make a continous flow of a software project in the industry.This numeric value show the minimum required resource for maximum productivity

2) *Time graph for Design phase:*In figure11,this graph is generated for Design phase in incremental model.The required optimal resource for desginer is 6.22 which make a continous flow of a software project in the industry.This numeric value show the minimum required resource for maximum productivity.

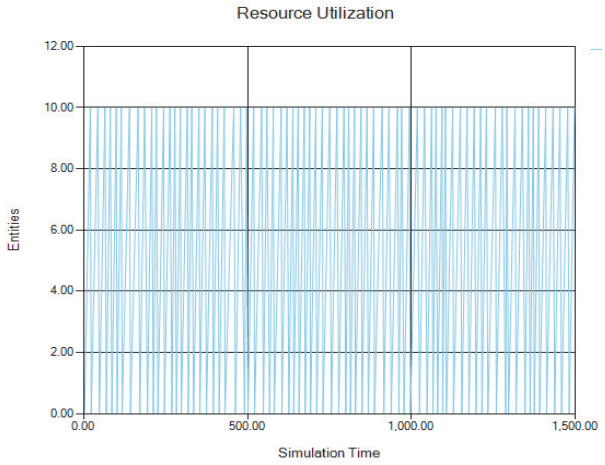

Fig 10 Utilization of business Analyst

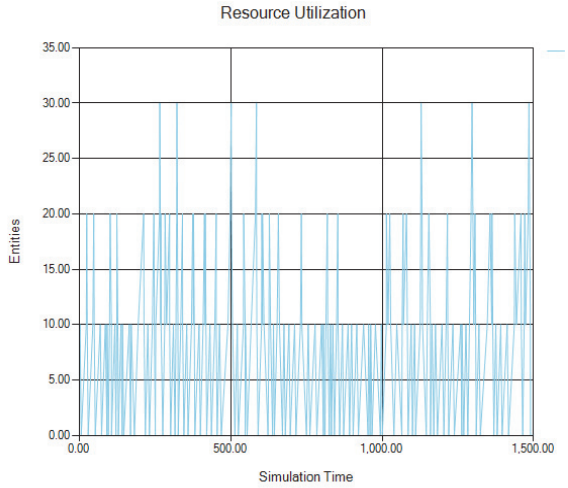

Fig 11Utilization of Designer

3) Time graph for Implementation phase: In figure 12,this graph is generated for implementation phase in incremental model.The required optimal resource for programmer is 70.03 which make a continous flow of a software project in the industry.This numeric value show the minimum required resource for maximum productivity.

*4) Time graph for testing phase***:** In figure 13,this graph is generated for testing phase in incremental model.The required optimal resource for tester is 79.89 which make a continous flow of a software project in the industry.This numeric value show the minimum required resource for maximum productivity.

5) Time graph for maintenance phase: In figure 14,this graph is generated for maintenance phase in incremental model.The required optimal resource for maintenance man is 5.24 which make a continous flow of a software project in the industry.This numeric value show the minimum required resource for maximum productivity

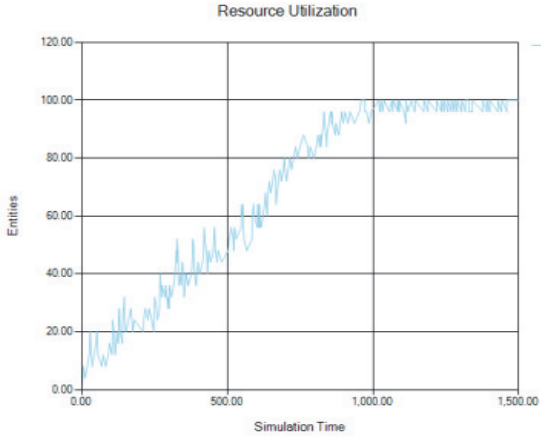

Fig12 Utilization of programmer

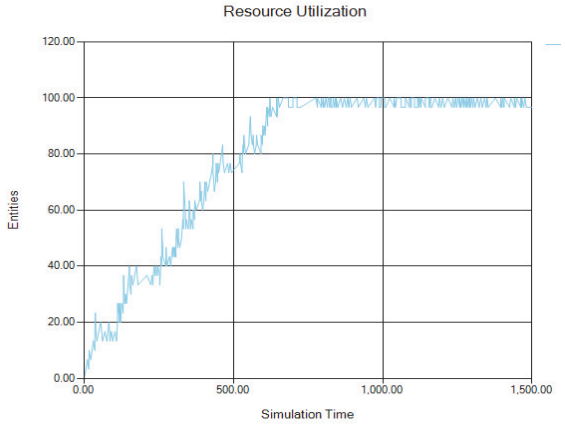

Fig 13 Utilization of Tester

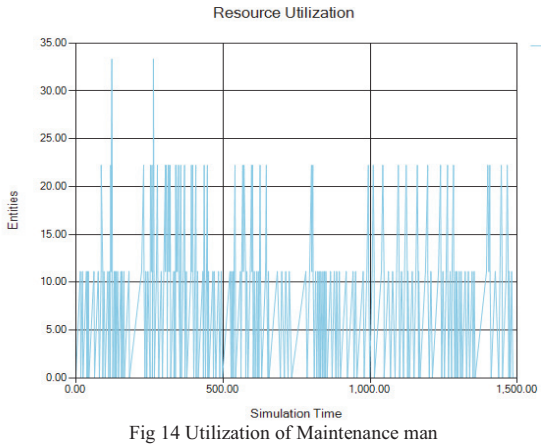

Fig 14 Utilization of Maintenance man

Table below shows optimized resource of iterative waterfall and incremental model. It is shown that the required optimal resource is more in the last three phases of incremental model in comparison to the iterative waterfall model.

TABLE 2 COMPARISONS FOR OPTIMIZED RESOURCES BETWEEN ITERATIVE WATERFALL AND INCREMENTAL MODEL

Resource	Iterative Waterfall model	Incremental model
Business Analyst	8.60	2.27
Designers	20.10	6.22
Programmers	33.81	70.03
Testers	13.32	79.89
Maintenance man	5.78	5.24

VII CONCLUSION

This paper have proposed simulation models to compare iterative waterfall and incremental software life cycle model for optimizing resource using the Simphony.NET simulation tool. All the phases consisting in the iterative waterfall and incremental model are simulated. Our aim is to determine the optimal number of resource required to produce a project within allocated time, budget and quality product.

As future work, we shall be simulating spiral and iterative incremental model to determine which Software process is more appropriate for software industry.

REFERENCES

[1] NABIL MOHAMMED ALI MUNASSAR AND A.GOVARDHAN, "*A COMPARISON BETWEEN FIVE MODELS OF SOFTWARE ENGINEERING*" IJCSI INTERNATIONAL JOURNAL OF COMPUTER SCIENCEISSUES, VOL7, ISSUE 5, SEP, 2010

[2]PRATAP K.J MOHAPATRA *" SOFTWARE ENGINEERING: LIFE CYCLE APPROACH*",ISBN 978-81-24-2721-9,1TH ED., NAIL, 2010

[3]Roger S.Pressman,"*Software engineering: A practitioner approach*",ISBN 0-07-365578-3,5th ed., TMH, 2001

[4] Stephen R. Schach,"*Object oriented software engineering*",ISBN 978-0-07-352333-0 ,1th ed,THM,2008

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

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

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

[8]CTG.MFA-003,"A survery of system Development process Models",Models fo action projects:Developing and preservation,center for technology in Government University at Albany/Suny.1998.

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

[10]Microsoft .NET Framework 1.1Avaliable[online]:www.microsoft.com/en us/download/detalils.aspx?id=26

[11] Simphony.NET(2005),UniversityofAlberta,[Online].Availabl:http:// irconstruction.ualberta.ca/html/research/software/simphony.net.html

[12] Rupa Mahanti, M.S. Neogi and Vandana Bhattacherjee,**"** *Factors Affecting the Choice of SoftwareLifeCycle Models in the Software Industry-An Empirical Study",Journal of* ComputerScience: 1253-1262, ISSN 1549-3636,© 2012 Science Publications

[13]Youssef Bassil, "*A Simulation Model for the Waterfall Software Development Life cycle*", International Journal of Engineering &Technology, ISSN: 2049-3444, Vol.2, No.5, 2012

[14] Extreme Chaos (2001), Standish Group, [Online]. Available: http://standishgroup.com/ Sample research/extreme_chaos.pdf

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

[16] Christian walck "*handbook on statistical distribution* "university of stockholem, 2007

[17] Balci,O ," Guidelines for successful simulation studies",O Proceeding of the simulation the Simulation Conference pp.25-32,New Orleans,LA,1990.