DOI: http://dx.doi.org/10.26483/ijarcs.v8i7.4136

ISSN No. 0976-5697

Volume 8, No. 7, July – August 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

# A SIMULATION MODEL FOR INCREMENTAL SOFTWARE DEVELOPMENT LIFE CYCLE MODEL

Amninder Singh
Research Scholar
UIET Panjab University
Chandigarh, India

Puneet Jai Kaur
Assistant Professor
UIET Panjab University
Chandigarh,India

*Abstract:* Software Development Life Cycle, SDLC for short, is a very much characterized, organized sequence of phases in software engineering to add to the proposed software product [12]. SDLC gives a progression of ventures to be taken after to plan and add to a software product effectively. This research deals with a fundamental and significant issue in computer world. It is concerned with the software management processes that study the area of software development through the development models, which are known as software development life cycle (SDLC).This paper also describes the software development life cycle model of software used in the field of software development. These models are categorized two types: traditional Models and contemporary models. Waterfall model, incremental model, spiral model and the V-shaped model are comes under traditional model. The contemporary model consist agile model and Extreme programming model. These are widely used in industry. In this paper, we are proposing a simulation model for iterative process using .NET simulation tool. The purpose of the research is to determine how to get the maximum productivity by using less number of resources, in minimum time.

***Keywords:*** Software Engineering; SDLC; Incremental model; Computer Simulation; .NET;

## I. INTRODUCTION

The process of setting up a computer software and information systems is determined by the different ways of development methodologies. To plan, manage, and control the process of developing information system software development methodology framework has been used. A software development technique, officially known as SDLC, mainly used in systems engineering, software engineering, mechanical engineering, computer science, computational science and engineering applications, and some engineering and industrial fields [1]. SDLC is a structure based on development of a software product. It is a group of system development life cycle (SDLC) [2]. As the need of computers have increases day by day. Thus, Computers find their applications in different areas like in banking, medical field, educational field etc. All these area requires software according to their applications and need. Hardware alone is not fulfilling all the needs to do some useful work. So, Software must be used in addition with the hardware. To provide quality of product an engineering field known as software engineering is used. Software engineer will provide the quality of product, reliability of the product and estimated budget. To provide all these needs every company has go through different process. Therefore, software model is used to provide a method for developing a software product.

SDLC is essential because it tells the software developers that what they exactly need from software development program. SDLC ha break the whole cycle of software development and thus the SD (Software development) has been evaluated properly and helps the programmers to evaluate the program properly. SDLC also provide a spare time when the software will be available for use. SDLC works in steps, So if any of the step get missed whole outcome get effected.

The aim of the research work is to identify the optimal number of resources in minimum time. The size of the company varies according to the number of employees that works on various projects (Small, medium and large).We are using the resources effectively so that quality of work and productivity of the company rises up.

This paper is organized as follows: The brief review of SDLC was given in section 1. The description about SDLC model and related work was given in section 2 and section 3 respectively. Section 4 presents the simulation model and experimental results. Finally, the conclusion is made based on the results and discussion in section 5.

## II. SDLC MODELS

Software development life cycle (SDLC) is a process used by many industries to design, develop and test the quality of the software product. The main aim of the SDLC is to provide a high quality product. According to IEEE standard glossary of software engineering terminology, the software life cycle is defined as the time between when a software product is conceived and when the product is no longer available for use. ‘Software life cycle typically includes the following activities:

- Requirements Analysis
- Specification
- Software architecture
- Implementation
- Testing
- Documentation
- Training and Support
- Maintenance

The requirement gathering and analysis phase helps to understand the problem. A plan has been made to solve this problem in the first stage that is at the designing stage. This planned is implemented during coding. Solution methods have been made for different test cases. These phases have been followed by the deployment and maintenance stage. These various development stages has been designed using software product known as Software Life cycle model [4].The above actions form part of framework activities and are performed in all software development project [3].

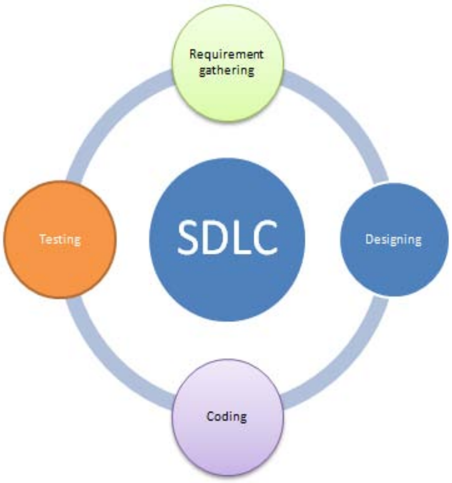


Figure 1: Cycle of SDLC

SDLC is very necessary for the success of software project, a good software engineer should have the sufficient knowledge and good experience to select one model than another based on the project situation. Therefore, it may be necessary to select the right SDLC model according to the exact concerns and requirements of the project. In this section, we will survey the different types of models and the merits and demerits of each one and when to use them [5]. At the last phase when the product is generated this is known as software product. Mainly SDLC models are categorized into two types:

**A. *Traditional models***

1. Waterfall model
2. Incremental model
3. Spiral model
4. V-shaped model
5. RAD model

**B. *Contemporary models***

1. Agile model
2. Extreme programming model

### *A. Traditional Models*

These models are characterized according to their linear nature. Some of the traditional models are discussed below:

#### *1) Waterfall Model*

It is also known as sequential SDLC model because in this different phases are carried out in a sequence. In this model, development has been done from upward to downward stream in sequence. Following is the figure showing the sequential process. In this the process is downward like a waterfall. Hence, if first condition is verified only then the next process has been executed. If any change occurs in the back process it does not go back to handle that process.

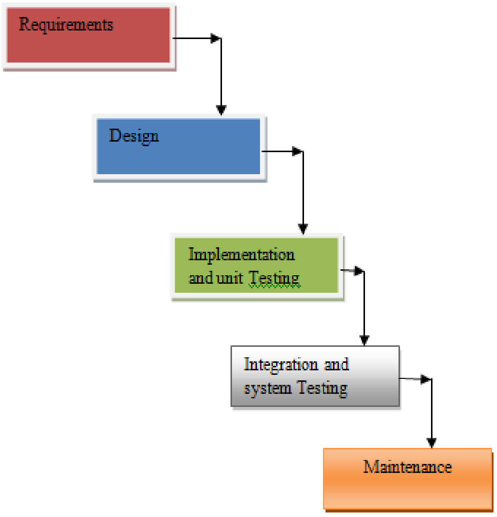

Requirements
Design
Implementation and unit Testing
Integration and system Testing
Maintenance

Figure 2: waterfall models [6]

**The successive stages in Waterfall model are:**

a) *Requirement Gathering:* All possible prerequisites of the framework to be created are caught in this stage and reported in a necessity particular document.

b) *System Design:* The necessity particulars from first stage are considered over in this stage and framework configuration is arranged. Framework Design helps in indicating equipment and framework prerequisites furthermore helps in characterizing general framework architecture.

c) *Implementation:* With inputs from framework design, the framework is initially grown in little projects called units, which are incorporated in the following stage. Every unit is created and tried for its usefulness which is alluded to as Unit Testing.

d) *Integration and Testing:* All the units grew in the execution stage are coordinated into a framework subsequent to testing of every unit. Post combination the whole framework is tried for any issues and failures.

e) *Deployment of framework:* Once the functional and non-functional testing is done, the product is sent in the client environment or discharged into the business.

f) *Maintenance:* There are a few issues which come up in the customer environment. To settle those issues patches are released. Additionally to improve the item some better forms are released. Maintenance is done to convey these adjustments in the client environment.

#### *2) Incremental Model*

To develop application software product a standard model known as Incremental model is used. In these model different processes like designing, testing and implementation have been done until; we will get the required product. This model is used to consider the customers changes as per their requirements. The waterfall model does not consider the previous changes[7].

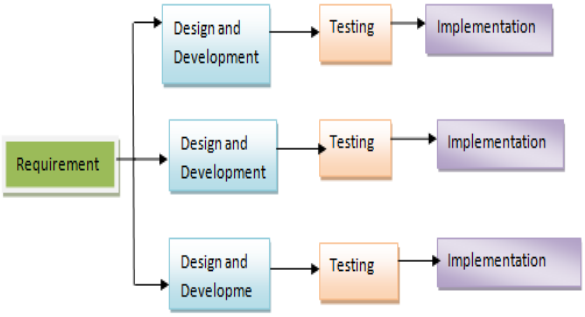


Figure 3: Incremental model

a) *Requirement phase:* This is the first step in the incremental model. A product analyzer expert identifies the basic requirements. After analyzing the product demand by the customers team members are ready to make document and categorized the working of the system. In documentation analyst are decided how to complete the task in step by step.

b) *Design and development phase:* In this phase design of the system and development process has been done effectively. Whenever any new product or any changes in the product are required, incremental model use design and development.

c) *Testing Phase:* Different testing methods are used in testing phase in order to check the behavior of the system response.

d) *Implementation phase***:** In this phase, the coding that has been developed by the developers has been implemented.

### 3) Spiral Model

It is a type of software development process that will combined elements of both design and prototyping stages. It will combine the merits of top down and bottom up concepts. Using spiral model the software's are developed in a series of iterations. Spiral models are divided into different activities. These activities demonstrate segments of the spiral path. In this model the team will perform their work in clock-wise direction. Spiral processes mainly includes: Requirement analysis phase, designing phase, coding phase, testing phase and maintenance phase [8].

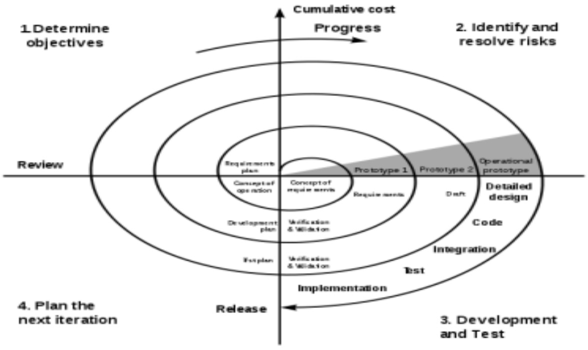


Figure 4: Spiral model [8]

### 4) *V-Shape Model*

It is similar to the water fall model. In this the execution of the process occurs sequentially. Before going to the next stage, previous stage must be completed successfully. This model required more testing than the waterfall model that is testing is done in every phase of the system.

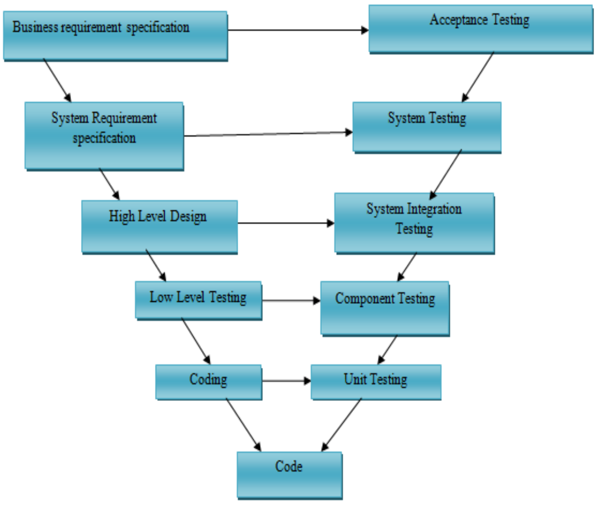


Figure 5: V- shaped model [9]

## B. *Contemporary models*

The contemporary software development models are based upon the principle of iteration of SDLC processes. A few contemporary models are discussed below:

### 1) *Agile Model*

This model was introduced by the agile team in 2001 using agile manifesto. In order to achieve customer's satisfaction, it focused on regularly delivery of the software using iterative and incremental development. The main aim of the agile model is discussed below:

- Incremental: Small software releases, accompanied by fast development cycles
- Co-operative: Closer customer-developer interaction
- Adaptive: flexible enough to adapt to last moment changes

Agile methods are lightweight processes that focus on short iterative cycles, direct customer and user involvement, incremental development and accommodating change even late in the development lifecycle. Also, these methods are value-driven; that is, they focus on maximizing the business value to the stakeholders involved. "Business value is something that delivers profit to the organization paying for the software in the form of an Increase in Revenue, an Avoidance of Costs. The main objective of the software development team is to produce working software at regular intervals. Agile methods advocate the adoption and use of iterative and incremental development approaches. Working software is used to measure progress and minimal documentation is produced.

### 2) *Extreme Programming Model*

This model is used for developing software in an flexible environment. Due to its flexible nature the cost of change in requirements during the later development phase decreases. XP provides a simple, precise and seemingly native principles and values to guide the software development process in four main stages of software development: planning, coding, design and testing (Figure 6). The main objective is to provide the needs of the customer when necessary. In addition, one of the main reasons for its success is the ability to accept changes at any time in the process of development

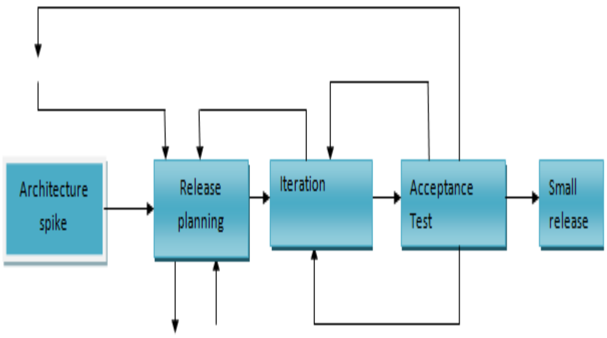
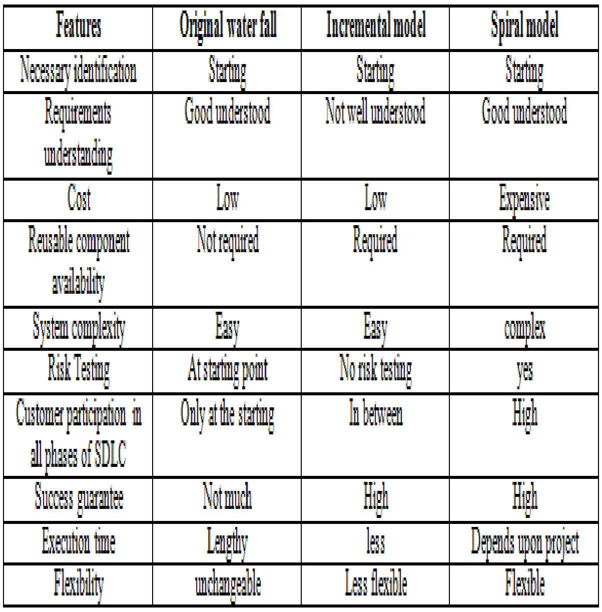


Figure 6: Extreme programming project [10]

## III. RELATED WORK

**Toussef basil [1]** proposed a simulation model for simulating the waterfall software development life cycle by using .NET simulation tool. Author provided the simulation results for all the entities like operational resources, employees, task and phase. The main aim of the project was to generate a project within the allotted time and budge. **Shika maheshwari and Dinesh Ch. Jain [2]** compared different software development models. Author has discussed different advantages and disadvantages of various software development models. **PK.Ragunath, S.Velmourougan[6]** proposed an hypothetical model, in this model author converted the SDLC model into three dimensional model which consist user, owner and developer. **Sonia Thind and Karambir [8]** proposed a spiral development process using .NET simulation tool. This tool helps the project manager to understand that how the productivity of a company can be increased using minimum number of resources. This has been done by keeping all the developers busy all the time, so that no time will be wasted and the idle time gets utilized. **A. Kaur and H. Monga [11]** auxiliary characteristics of capacity situated programming parts are investigated utilizing programming measurements and the measurements utilized are Cyclometric Complexity Using Mc Cabe's Measure, Halstead Software Science Indicator, Regularity Metric, Reuse recurrence metric, Coupling Metric. **A.Bakhshi, S.Bawa [12]** produces a product segment archive to store reusable component. **Dan Turk et al. [13]** identifies the limitations of agile software process. **Mumtaz Ahmad Khan et al. [14]** presented a method for choosing the SDLC model using AHP. Author recognized that it is necessary to improve the agile methods by interwining of decision making approaches. **reuven r. levary and chi y. lin** [15] described about the intelligent programmed tool designed to aid managers of software development projects in planning, managing and controlling the development process of medium to large scale projects. **Stewart Robinson [16]** provided an overview about discrete –event simulation in facilitation. Here DES (discrete event simulation) was used to facilitate stakeholders in a problem situation**. C. Y.Lin et al. [17]** presented a software process simulation (SEPS) model developed at JPL. This software used feedback principle in order to simulate the dynamic interaction among various SDLC. **Osman Balci [18]** provides gaudiness for conducting a successful simulation study. The author's takes 10 processes, 10 phases and 13 credibility assessment steps for simulating the SDLC. **Prakriti Trivedi and ashwani Sharma [19]** presented a model to solve the problems of assigning optimal resources for software life cycle model occurs in incremental and Incremental model Literature survey

Table1: Comparison of three models [2]



| Features | Original water fall | Incremental model | Spiral model |
|---|---|---|---|
| Necessary identification | Starting | Starting | Starting |
| Requirements understanding | Good understood | Not well understood | Good understood |
| Cost | Low | Low | Expensive |
| Reusable component availability | Not required | Required | Required |
| System complexity | Easy | Easy | complex |
| Risk Testing | At starting point | No risk testing | yes |
| Customer participation in all phases of SDLC | Only at the starting | In between | High |
| Success guarantee | Not much | High | High |
| Execution time | Lengthy | less | Depends upon project |
| Flexibility | unchangeable | Less flexible | Flexible |

Above table explains the features of models undertaken namely, Original Waterfall model, Incremental Model and Spiral Model. Primarily, the necessary identifications are taken as the features which are considered as the initial phase of any model. Next, the requirements needed for starting of any project/model etc is taking place. From the three models, original waterfall and Spiral model has good understanding of the requirements. According to Budget (cost) and execution time wise incremental model is best. Incremental model and spiral model components are reuse whereas in waterfall model components are used only one time. The spiral model designing is complex than the other two models. In waterfall model testing is done at the initial stage, in incremental model testing is done when required; in spiral model testing is done in iterative phase. Customers can interfere only at the beginning of the original waterfall model, in incremental model; customers can interfere at any stage of the model. Incremental model is highly successful than other two models. In waterfall model, we changes take place at the starting point, in incremental model and in spiral model changes take place at any stage.

## IV. THE SIMULATION MODEL

The research paper offers a simulation model, in which the various phase of iteration models like designing, testing and implementation has been performed[20]. The Incremental model is simulated by using the simulation tool known as .NET. The main aim of the simulation is to provide the resources for each project along with time constrain. For example, a small project is completed in 10 days, so, for completing small projects best possible employees should be assigned to small project. For medium project, there are 30 days allotted for the completion of the projects. Thus according to the requirement best possible employees should be allotted to the best, so that the project should be completed within time. For large project, the project will be completed within 50 days. So, by keeping ideal and busy resources minimum, bets possible employees are allocated to the project[21].

Following are the simulation process used for the proposed work:

Step 1: After completing the simulation, examine the output data obtained after simulation.

Step 2: After simulation check the changes occurs in the model.

Step3: Repeat the process until the desired results has been obtained.

The simulation processes of the Incremental model are listed below:

- The Incremental model is divided into independent sources
- Understand the concept and requirements of the iterative model.
- Define the resources for business analyst, designer, programmer, tester and maintenance man.
- Simulate every phase of the model and record the results.
- Repeat the process until the desired results has been obtained.

### A. *Assumptions and specifications*

Before designing and testing the iterative model, a number of assumptions have been taken into mind. The project has been divided into three parts small, medium and large projects. The probability function is given as:

$$F\{y(a,b,c)\}=\begin{cases} small & for\ 10 \geq y > 0 \\ medium & for\ 30 \geq y > 10 \\ Large & for\ 50 \geq y > 30 \end{cases} \quad (1)$$

Projects have been divided into three groups depending upon their complexity and scale: 60 % of the projects are small scale, 30% of the projects are medium scale and 10% of the projects are large scale[22].

Every project requires a group of Specialist employees (SE) and resources have been given to the projects depends upon their project. Some terms used for the projects are listed below:

Business analyst (BA)
Designer (D)
Programmer (P)
Tester (T)
Maintenance man (MM)

i. For small scale projects, we need 1 BA, 1 D, 2P, 2T and 1MM.
ii. For medium scale projects we need 2 BA,2 D, 4 P, 6T and 2 MM.
iii. For large scale projects we need 5 BA, 5 D, 10 P, 20 T and 5 MM.

There are some assumptions are taken for the completion of project[23]:

1. Small Project must be completed in 10 days
2. Medium Project Must be completed in 30 days
3. Large projects must be completed in 50 days according to a medium firm.

### B. *Simulation Results*

The simulation result was executed for 10 firms, having size small, medium and large. The range of employees varied from 100 to 1000. For small companies minimum numbers of employees are 300, for medium 500, for large companies minimum employees are 700.

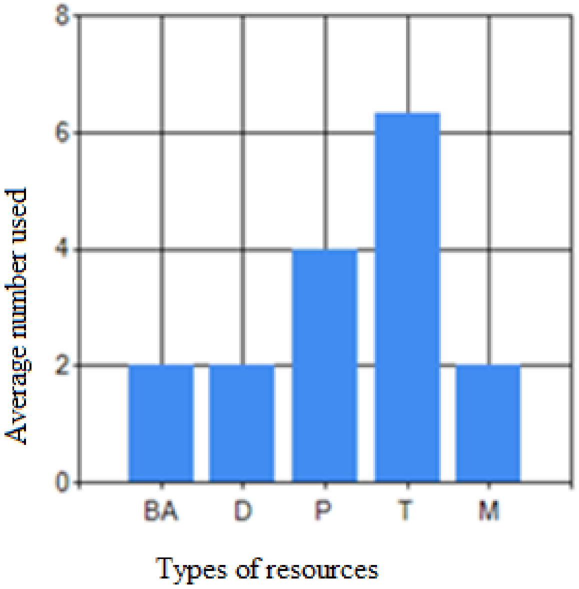


Figure 7: Average number of resources used in a medium firm

The above figure depicted that there are average number of 2 business analysts, 1 designer, 4 programmers, 6 Testers and 2 maintenance men are allotted for a medium firm for single project.

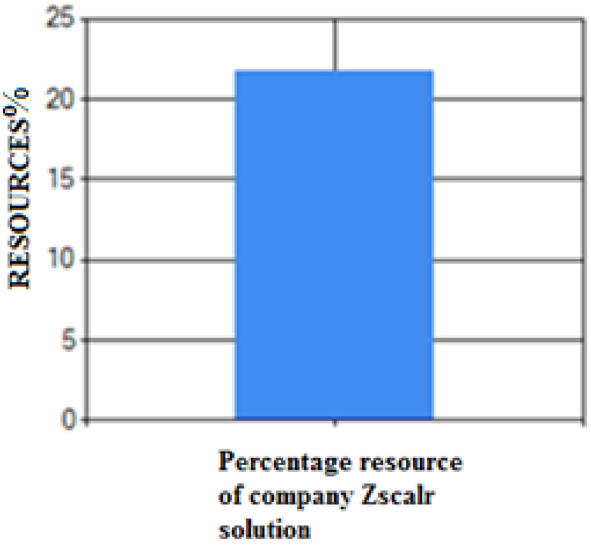
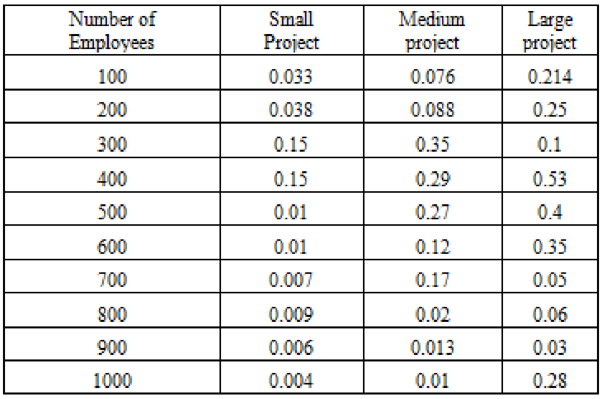


Figure 8: Average of resources used in medium firm

The above graph obtained for the average number of the total resources calculated for the medium firm. The percentage of resources obtained for the medium firm is approximately equal to 22% where total number of resources is 100.

Table 2: number of resources used (%)



| Number of Employees | Small Project | Medium project | Large project |
|---|---|---|---|
| 100 | 0.033 | 0.076 | 0.214 |
| 200 | 0.038 | 0.088 | 0.25 |
| 300 | 0.15 | 0.35 | 0.1 |
| 400 | 0.15 | 0.29 | 0.53 |
| 500 | 0.01 | 0.27 | 0.4 |
| 600 | 0.01 | 0.12 | 0.35 |
| 700 | 0.007 | 0.17 | 0.05 |
| 800 | 0.009 | 0.02 | 0.06 |
| 900 | 0.006 | 0.013 | 0.03 |
| 1000 | 0.004 | 0.01 | 0.28 |

The list of number of resources used for 10 firms according to small, medium and large projects is listed in table 2.

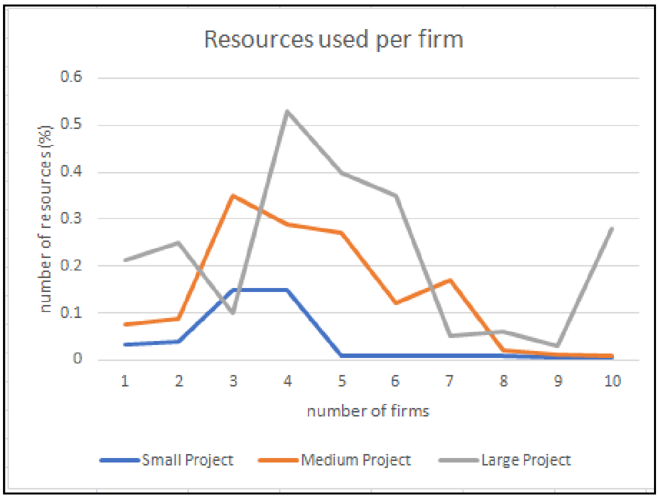


Figure 9: resources used per firm (Small, medium and large projects)

Total number of Firm = 10

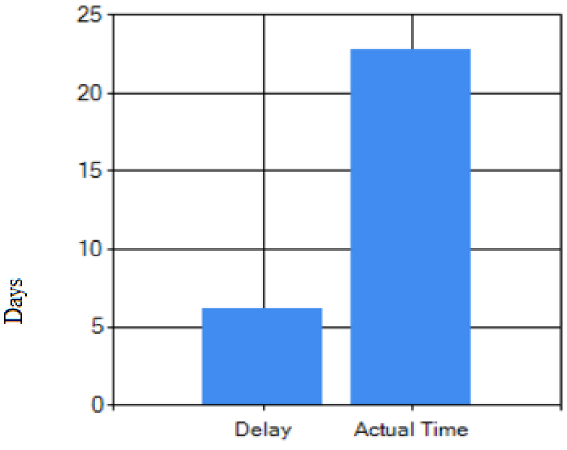


Figure 10: Delay in medium firm projects

The above figure depicted that, there is a delay of 6 days for completing the medium firm for 10 projects where 5 projects are completed before assumption (actual) time and 5 are completed after assumption (actual) time.

## V. CONCLUSION

In this paper we proposed simulation model for simulating the Incremental software development life cycle by using .NET tool. It consists all the entities of iteration model including firm's employees, time and various resources. The aim of the research work is to use optimal number of resources, and give output within the allotted period. The resources like employees are allocated after initialization of project and de-allocated after completion of projects so that the company works simultaneously on large number of projects. The simulation is varies according to 10 firms, which are working for different projects sizes like small, medium and large. The size of firms is varied in accordance with the number of employees. The results prove that limited number of employees worked on many projects in associative manner. In future, other SDLC like spiral can be used. This is used for large and critical projects, where the requirements' are complex.

## VI. REFERENCES

1. Y. Bassil, "A simulation model for the waterfall software development life cycle" , International Journal of Engineering & Technology, vol. 2, no. 5, 2012.
2. M. Shikha and J. Dinesh Ch, , "A Comparative Analysis of Different types of Models in Software Development Life Cycle," International Journal of advanced research in computer science and software engineering vol. 2, no. 5, 2012.
3. P. Sharma, D. Singh "Comparative Study of Various SDLC Models on Different Parameters." International Journal of Engineering Research Vol..4, no.4, 2015.
4. S. Kaur , "A Review of Software Development Life Cycle Models,"International journal of advanced research in computer science and software engineering, vol.5, pp3.54-361, ,2015.
5. https://melsatar.blog/2012/03/15/software-development-life-cycle-models-and-methodologies/
6. PK. Ragunath, S. Velmourougan , P. Davachelvan ,S. Kayalvizhi, R. Ravimohan† " Evolving a new model (SDLC Model-2010) for software development life cycle (SDLC)," International Journal of Computer Science and Network Security, vol. 10, no .1, 2010.
7. http://er.yuvayana.org/sdlc-incremental-model-design-phase-applications-advantages-and-disadvantages/
8. S. Thind, Karambir, "A Simulation Model for the Spiral Software Development Life Cycle," International Journal of Innovative Research in Computer and Communication Engineering, vol.3, 2015.
9. N. Munassar Mohammed Ali, and A. Govardhan, "A comparison between five models of software engineering," International Journal of Computer Science, vol. 7, 2010.
10. M. Hneif and S. Hock Ow, "Review of agile methologies in software development,"International Journal of Research and Reviews in Applied Sciences ,vol 1,no. 1, 2009.
11. S. Iffat Zahara, M. Ilyas, and T. Zia., "A study of comparative analysis of regression algorithms for reusability evaluation of object oriented based software components,"Open Source Systems and Technologies , International Conference on. IEEE, 2013.
12. A. Bakshi, and S. Bawa, " Development of a Software Repository for the Precise Search and Extract Retrieval of the Component.,"Diss. 2013.
13. D. Turk, R. France, and B. Rumpe., "Limitations of agile software processes," Third International Conference on Extreme Programming and Flexible Processes in Software Engineering, 2014.

14. M. Ahmad Khan, A. Parveen, and M. Sadiq, "A method for the selection of software development life cycle models using analytic hierarchy process," Issues and Challenges in Intelligent Computing Techniques, International Conference on. IEEE, 2014.
15. R. Reuven Levary, and Chi Y. Lin, "Modelling the software development process using an expert simulation system having fuzzy logic," Software: Practice and Experience vol 21, no. 2, 1991.
16. S. Robinson, "Soft with a hard centre: discrete-event simulation in facilitation.," Journal of the Operational Research Society ,vol. 52, no..8, 2001.
17. C. Y Lin, "Software-engineering process simulation model ." Journal of Systems and Software , vol 38, no. 3 1997.
18. P. Trivedi and A. Sharma, "A comparative study between iterative waterfall and incremental software development life cycle model for optimizing the resources using computer simulation," 2nd International Conference on Information Management in the Knowledge Economy, Chandigarh, pp. 188-194, 2013.
19. S. Robinson,"Modes of simulation practice: approaches to business and military simulation," Simulation Modelling Practice and Theory , vol. 10, no..8 ,2002.
20. M.V. Satish Kumar, ,K. Amarnath, G. Sanjeev, M. Pradeep Kumar. "Process Simulation and Process Optimization for Driving License Process by Using Process Simulator," International Journal of Innovative Technology and Research, vol.4, no.1, 2016.
21. I. Singh, and A. Singh, "Gray testing support in software repository with keyword based extraction ," Advances in Computing, Communications and Informatics, International Conference on. IEEE, 2014.
22. J. Anubha, C. Swati , and P. Tiwari, "Relevance of Genetic Algorithm Strategies in Query Optimization in Information Retrieval,"International Journal of Computer Science & Information Technologies, vol. 5, no. 4, 2014.
23. W. Royce, "Managing the development of large software systems: concepts and techniques," presented at the Proceedings of the 9th international conference on Software Engineering, Monterey, California, United States, 1987.