ISSN 2229-5518

A Hybrid Model for the Enhancement in Software Effort Estimation

Tajinder Kaur[1], Jaspreet Singh[2]
[1]Post Graduate Student, Department of Computer Science & Engineering (Software Engineering)
Chandigarh University, Mohali, Punjab,India
er.tajinder1991@gmail.com
[2] Assistant Professor, Department of Computer Science & Engineering
Chandigarh University, Mohali, Punjab,India
jaspreet.rajal@gmail.com

Abstract— Software cost estimation plays a vital role in software engineering as the success or failure of project entirely depends on it. Accurately estimating the cost of software projects is one of the most desired capabilities in software development organizations. An estimate is not really a prediction, it is a management goal. Measurement of work involved in completing a project is called size of the project.Effort and time required to develop the software can be computed by estimating the project size.Inaccurate cost estimation may lead to project failure,huge overruns and performance compromises as a consequence.In this paper, a hybrid parametric model has been proposed with size estimation model which helps to determine a set of homogeneous projects by using a technique derived from estimation by analogy.

Index Terms— COCOMO,Estimation,Function Point, Prediction,SLIM.

1 INTRODUCTION

Even though demand of software system is increasing in all aspects of human life, development of software projects are hysterically noted with delays, high cost and errors. Inaccurate estimation of resources is major factor of failure in software projects[1].Software estimation draws enormous attention from academicians and specialists. Software estimation is the mechanism of predicting cost, effort and duration that are required to develop the software [2]. Estimator often depends on number of pragmatism to generate software estimations[3]. Exceeded budgets, function that are not developed completely, low quality and partial completion of the project are some of the major factors that results in underestimation or over estimation of the software cost [2].Estimation of cost and size of the software project are the biggest challenge. A project budget, schedule and size of development team are directly dependent on the estimation. The process of software development effort estimation in which estimator predict the amount of effort in term of person-hour or monthly to maintain software are based on uncertain, noisy input and incomplete project plan, budgets, pricing processes and investment [15].

1.1 Software Cost Estimation

Software cost estimation plays a vital role in software engineering as the success or failure of project entirely depends on it.Cost estimation's deliverables like staff requirements, schedule and effort are important chunk of information for formation and execution of the project. They provide inputs for project request and proposal, project planning, control, budget, progress monitoring & staff allocation. Illogical and uncertain estimates are the root cause of project failure. So, the capability of the system is to find out correct time and cost of the software is very crucial for the progress of the system.The software engineering community puts enormous effort while buildings models in order to comfort estimators to provide accurate cost estimates for software projects. COCOMO, SLIM, SEER-SEM and Price- S are some of the estimation models that have been proposed and used in the last three decades.

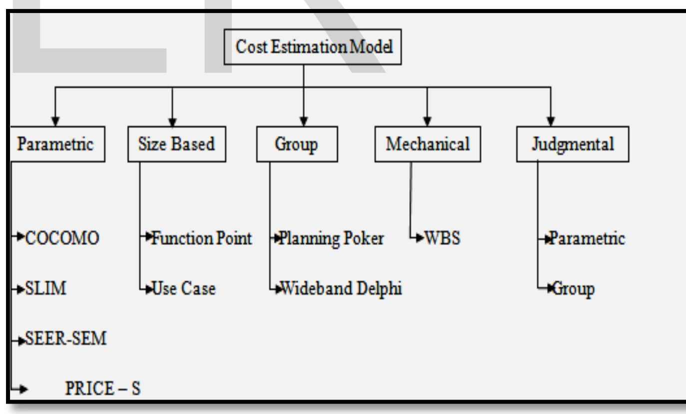

Fig.1.1 Estimation Models

1.2 Models for Cost Estimation

There are various cost estimation models.It is classified as:

1.2.1 Parametric Model

It is an estimation technique that applies on one or more cost estimating affair and combines mathematical relationships and logic. It describes variables that provide numerical estimates for vital input variables that influence the effort or time used in development. COCOMO, SLIM,PRICE-S models come under it[19].

1.2.2 Size Based Estimation Model

The focus of estimation is on identifying the parameters that provides the size which is key focus for the tasks in the project. Function Point and Use Case models come under it.

ISSN 2229-5518

1.2.3 Group Based Estimation Model

It is as accurate as model-based effort estimation. Unstable relationships and information that are important are excluded from this model and for such cases model may use expert estimation as suggested. It requires experts with relevant experience[16].

1.2.4 Mechanical Estimation Model

This includes average of an analogy based and a Work breakdown structure based effort estimation.

1.2.5 Judge Based Estimation Model

This is combination of parametric model and group estimation model.

1.3 COCOMO Model

It is a mechanism for assessing the cost of software. It has three levels:

1.3.1 Basic COCOMO

It measures effort and cost as a function of program size. Its three models are:

Organic Model

It is simple & includes small experienced teams. The DSI are typically in a small amount that is under 50,000 and planned software is not considered innovative.

Semidetached Model

It is much difficult than the organic mode and team members have mixed levels of experience. Approximately, 300,000 DSI-are required by the software. It combines the characteristics of both modes.

Embedded Model

It is highly complex form of hardware & software.Regulations and operating procedures are dealt by this mode.

1.3.2 Intermediate COCOMO

A set of "cost drivers" are used to extend the basic model for measuring the effort. Valuation of personnel or hardware is assessed through this mode.

1.3.3 Detailed COCOMO

An extension of the Intermediate model that adds effort multipliers to determine the effect of cost drivers on each phase.

Table 1: Comparison of COCOMO1 & COCOMO2

COCOMO1	COCOMO2
It is a basic model.	It is an extensionof basic model.
It follows waterfall model.	It follows three phases concept.
There are 15 cost drivers.	There are 17 drivers present in this model.
It consists of 63 data points.	It consists of 61 data points.
It follows reengineering concept.	It follows software reusability.
It is measured in KDSI.	It is measured in KLOC.

2 Literature Review

D.Toka,O.Turetkan,"Accuracy of Contemporary Parametric Software Estimation Models" analyzed the accuracy of parametric software estimation models. In the paper,authors compared four parametric software estimation models in terms of effort and duration prediction accuracy.51 real project data is used to analyze the abilities of the models which is compared with the actual effort and duration values. The results of the models that are investigated are par on accuracy. The future work suggested by authors can be considered as incorporating historical data for adjustment purpose to have more insight into capabilities and strength of these methods and tools[4].

H.Rastogi, MishaKakkar,"A Survey on Software Effort Estimation Techniques",reviewed techniques and models of effort estimation. Comparison among several approaches is being done and the techniques that produces the most accurate result serves as a measure of selection.Every technique has its own merits and demerits. There is no single technique that can run away from all the shortcomings and can be globally accepted, so the future work suggested in the paper is hybridization of several approaches as an alternative to produce realistic estimation[5].

Poonam Pandey,"Analysis of the Techniques for Software Cost Estimation"analyzed the algorithmic techniques.As stated in the paper that there is no formal rule of thumb for determining the actual effort which is required for completing a project.The most important thing in the estimation is the datasets of the current and future projects which are required during evaluation of the estimation method. Not even a single factor that affects the project cost and development should be ignored while estimating the software development cost[6].

T.Matijevic,Ivana Ognjanovic et. al, "Enhancement of Software Projects Function Point Analysis Based on Conditional Non-Functional Judgment"presented the possible extension of function point analysis, which is widely used Functional Scaling Methods when it is required to enhance the processing of users non-functional requests. Different kinds of input parameter is used for the estimation of the complexity of software products. Different types of prioritization methods are also analyzed. The limitation of the approach is that as the number of requirements increases, it becomes harder for the user to select best tactics with inability to handle conditional requirements[7].

Chetan Nagar and Anurag Dixit, "Efforts estimation by combining the use case point and COCOMO",combines the Use Case point and COCOMO. They forecast the Line of Code with the help of Use Cases. Use Case used in the method must be more specific not more generalized.Use Cases gained wide popularity in software effort estimation. Results obtained using use cases are widely applicable. A strong monitoring policy is always required to make estimation as a success. A checklist is followed with the date of completion. If work is not done on the time some necessary action must be taken to compensate the deviation. To estimate the KLOC, divide the project into module and module into the further sub modules until it is possible to estimate the KLOC. Use Case Point shows the functional requirement of the system .So it is one of the good way of estimation[8].

ISSN 2229-5518

AdanmaC. Eberendu,“Software Project Cost Estimation: Issues, Problems and Possible Solutions”proposed hybrid model to enhance the accuracy of the estimation techniques. It is concluded that no single technique is sufficient that can do away with all the shortcomings. It is important to determine how much effort is required to complete the software project on-time. Hence, hybridization of more than one technique can provide more accurate estimation that can be helpful to avoid over-estimation or under-estimation of effort[9].

3 Function Point Based Estimation Model and COCOMO Model

Software engineering process or development process is the most effective method to analysis statistical data on the basis of history data. The ability of the system is to find out accurate time and cost of the software is very necessary for the success of the system. So there are different ways and different types of methods to estimate the cost and time of the software.

3.1 Function Point Based Estimation Model

This is a top down method devised by Allan Albrecht.
Albrecht was investigating on programming productivity and needed to quantify the functional size of programs independently of their programming languages. The basis of function point analysis is that information system comprises five major component or ‘external user types’ in Albrecht terminology that are benefited to the user.

a) External input types are input transactions which update internal computer files.
b) External output types are transactions where data is output to the user.
c) External inquiry types are transactions initiated by the user which provide information but don’t update the internal files.
d) Logical internal file types are standing files used by the system.
e) External interface file types allow output and input that may pass to and from other computer.

Function point analysis recognizes that the effort required to implement a computer based information system relates not just to the number and complexity of the features but also to the operational environment of the system[9].
Unadjusted function point: The sum of all occurrences and all the occurrences are computed by multiplying each function count with a weighing and then adding all the values.

$$UFP = FC1 * W1 + FC2 * W2 + \ldots\ldots\ldots + FCn * Wn$$

The constant value in it the equation and the weighing factor are determined empirically:

$$VAF = \text{total degree of influence} * 1\% + .65$$

On the whole,

$$FP = UFP * VAF$$

Function point(FP)computes the following important metrics:

Productivity: FP/person-month
Quality: Defects/FP
Cost: Rupees/FP
Documentation: Pages of documentation per FP.

3.2 COCOMO Model

COCOMO (Constructive Cost Estimation Model) was proposed by Boehm. It is very effective and oldest model for cost estimation. It is independent model which is well documented and cannot be depended upon any software vendor [22]. In COCOMO model line of code is estimated. In this model,it is understood that the complexity of the system is due to its openness nature[8][23].
In COCOMO model, there are three basic levels [24]:

I. Basic Cocomo
II. Intermediate Cocomo
III. Detailed Cocomo

I. Basic COCOMO:

As a function of program size, it computes software development effort and cost. It is static and single valued model.
In Basic COCOMO there are three modes:

a) Organic Mode
b) Semidetached Mode
c) Embedded Mode

*a) Organic Mode:*In this mode development team is small and consist of experienced persons. Here projects are not complicated. It requires less rigid requirements.
*b) Semidetached Mode:*In this mode,people are more experienced than organic level. This mode is more complicated than organic mode so complexity is more. It has characteristics of both modes organic and embedded. It requires rigid requirements.
*c) Embedded Mode:*In this mode software and hardware are complexly joined. It requires set of rigid requirements.
It may be combination of organic and semi-detached projects.

II. Intermediate COCOMO:

It is an addition to the basic model by adding a set of cost drivers that computes software development effort. In this 15 cost drivers are used to find out cost estimation of projects rated from very low to very high.

III. Detailed COCOMO

It is an extension of intermediate COCOMO. In this cost driver is added to effort multiplier at each phase to calculate the cost drivers. It uses different multiplier for each cost attribute. COCOMO 1 is also known as COCOMO 81.

3.2.1 Limitations of COCOMO

There are some limitations of this model which are as follows:
1. COCOMO starts estimation from the design phase and continues till the end of integration phase of cost and schedule of the project. A separate estimation model should be used for remaining phase.
2. Assumptions made at the starting in this model may vary as time progresses in developing the project. It is not a realistic perfect model.
3. A new estimation may show over budget or under budget for the project when to revise the cost of the project. This may lead to a partial development of the system.

ISSN 2229-5518

4 Proposed Methodology

Effort estimation means to estimate the efforts according to prospects stakeholders before project is being implemented. To measure the project size, usually in lines of code or equivalent is used. Software is a product without physical existence and its main cost is the design and development of the product. The effort is measured in term of man-month or year. There are many technique that estimate the efforts of software development but no single technique is sufficient that run away all the shortcomings. Thus,it is suggested that the models should be used in pairs to estimate the effort accurately. An approach is to combine the two or more techniques for the effort estimation of the software project.

Therefore, proposal is to peform Merge estimation technique. Firstly, estimate the size of the product to be developed is obtained through using function point. Function point analysis will be used to estimate KLOC value. As in function point analysis five components are used.i.e.: external input to the project, external output of the project, external query, internal logic files and external interface files. These all components are rated on the basis of their involvement in the project.

From the involvement of the components,generate complexity of the project. Convert the size estimates to cost estimation by combining the two or more techniques of the effort estimation. Then,compare techniques on the basis of their strengths and weaknesses. In this work, a formula can be derived that will provide KLOC value on the basis of number of adjusted and unadjusted functions in the project.

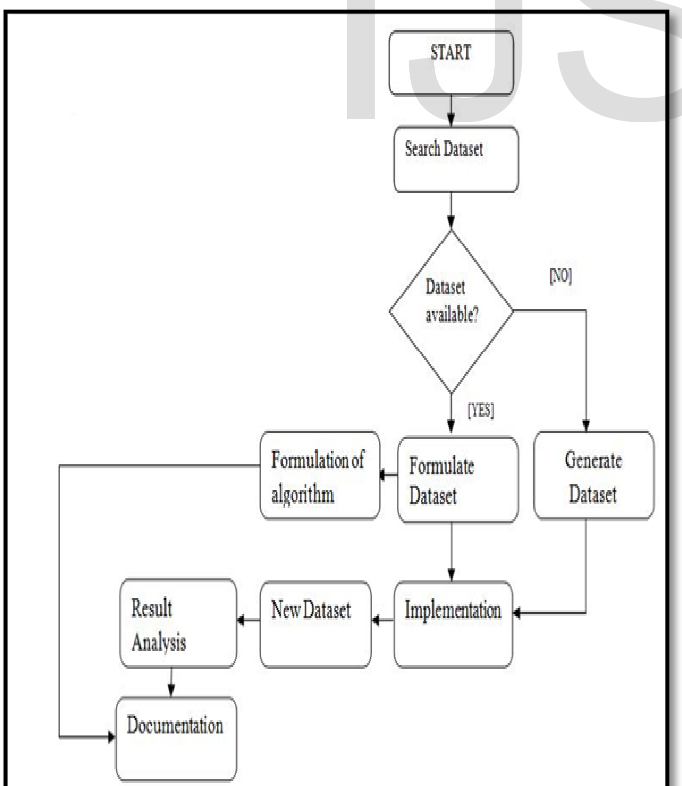

Fig.1.2 Flowchart of Methodology

COCOMO model is based upon the KLOC lines. But complexity increase as the lines of code increase. So it is complex to approximation the effort with more complexity. Function point estimation model is based upon functional size to estimate efforts. So to reduce complexity of COCOMO model in the proposed work, merge SLIM and function point model and enhance accuracy with the help of functional size as compared to KLOC in COCOMO model for effort estimation.

Effort derived from the proposed model calculates effort which is equal to

[((a+b)/b*0.1+2.5)*Size*0.99]1.266.

Where "a" is the sum of cost drives in the set PG i.e. cost drivers which is present in the Pessimistic Group,

"b" is the sum of cost drives in the set OG i.e. the cost drivers which is present in the Optimistic Group,

1.266 is Empirical Exponential Constant,

0.99 is Empirical Domain Constant 0.1

2.5 is Empirical Adjustment Factor.

5 Results & Discussion

The proposed work has been implemented in MATLAB. MATLAB is a numerical tool for manipulating matrix, implementing algorithms etc. It is the most widely used tool by the programmers in software industry as it can interface with any other language.

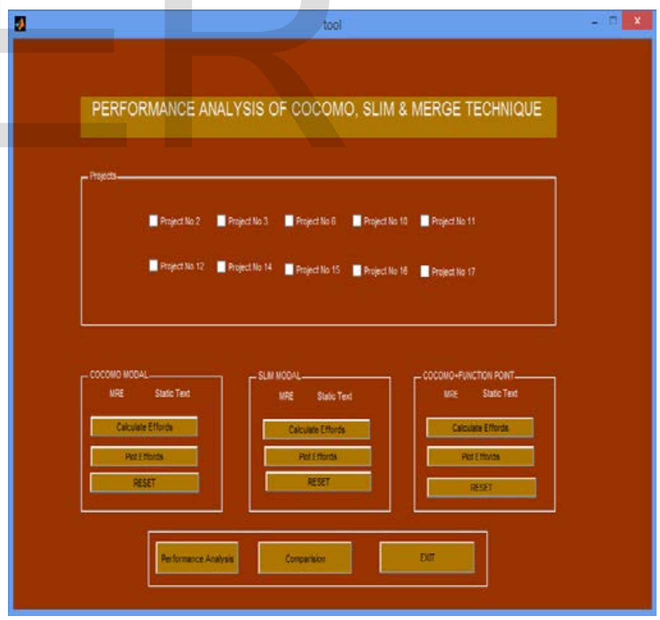

Fig 1.3 Tool default view

As illustrated in figure 1.3, the tool is developed to compare the performance of various cost estimation modals. These models are COCOMO modal, SLIM modal and hybrid modal which is the combination of COCOMO and Function point.

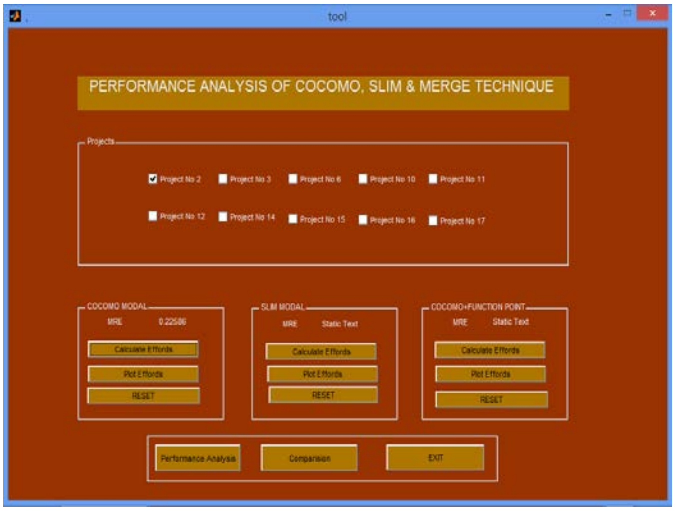

Fig 1.4 Performance of COCOMO Model

As illustrated in figure 1.4, the COCOMO model, SLIM model and hybrid models are compared in terms of MRE value. In this figure, the project1 is selected and using the CO-COMO model, its MRE value is 0.22586.

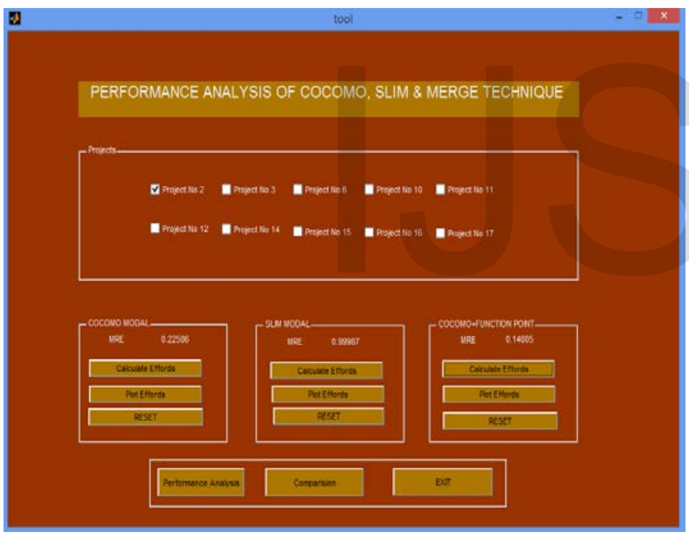

Fig 1.5 Performance of COCOMO+FUNCTION model

As illustrated in figure 1.5, the COCOMO+FUNCTION modal, SLIM modal and hybrid modals are compared in terms of MRE value. In this figure, the project1 is selected and using the COCOMO+FUNCTION modal its MRE value is 0.14805.
Efforts of each model can be calculated for different types of projects and graphs can be plotted in terms of MRE values.

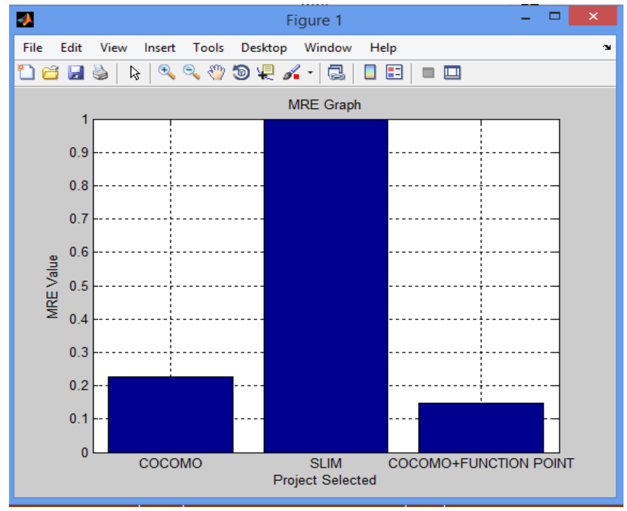

Fig 1.6 Comparison of three models

As illustrated in figure 1.6, the three models are compared on project 16, the comparison shows that higher MRE value is of SLIM modal, the second higher is of COCOMO modal and third higher value of MRE is of COCOMO+FUNCTION POINT modal. This graphs proves the efficiency of the devel-oped model.

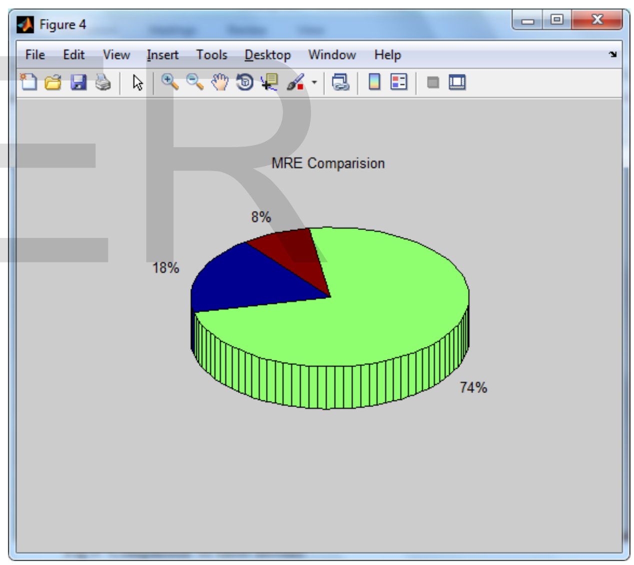

Fig 1.7 Pie chart comparison of three models

As shown in figure 1.7, the three models compared in terms of error rate on project 16. The SLIM modal had higher error rate i.e 74 percent. COCOMO modal is of 18 percent.
The COCOMO+FUNCTION POINT modal had least error rate i.e 8 Percent.

Table 2: Comparison Of Various Models On The Basis Of Different Projects

PROJECT No	COCOMO MRE	SLIM MRE	HYBRID MRE
Project 2	0.22586	0.99987	0.14805
Project 3	0.68095	0.99995	0.68305
Project 6	0.0255	0.9994	0.0069
Project 10	0.30394	0.99982	0.16279
Project 11	0.99734	0.99991	0.41309

6 Conclusion & Future Scope

Software estimation invloves predicting cost, effort and duration that are required to develop the software.Exceeded budgets, functions that are not developed completely, low quality and partial completion of the project are some of the major factors that results in underestimation or overestimation of the software cost. It may leads to project failures.Estimation of cost and size of the software project are the biggest challenge. In this paper, a hybrid parametric model has been proposed to deliver accurate & failure free estimates and that to within a specified period of time. This will in turn lead to a reduction of the number of IT-project failures and make better use of scarce financial and human resources. The proposed work offers important perspectives on the role of effort estimation in the development process and shows how effort estimation directly improves software quality and output efficiency. The performance of the developed method is tested on NASA software project data and results are compared with many estimation model like SLIM.The result shows that the proposed method has lowest MRE than other models.

SLIM modal is also the cost estimation modal which works on KLOC and cost driver values. It has higher error rate as compared to other models like COCOMO & Function Point. In future, proposed technique can be improved to enhance the performance of SLIM modal using function point estimation.

Acknowledgment

I would like to express my gratitude to Assistant Professor Mr.Jaspreet Singh, my research work supervisor, for his valuable and constructive suggestions during the planning and development of this research work.His unstined support and encouragement for the preparation of the manuscript has been very much appreciated.

References

[1] R. N. Charette, “Why Software Fails?” IEEE Spectrum, Sept.2005, 2005.

[2] H. Leung and Z. Fan, “Software Cost Estimation,” pp. 1–14, 2002.

[3] G. Stark, “A Comparison of Parametric Software Estimation Models Using Real Project Data,” Crosstalk, no. Jan/Feb 2011.

[4] DeryaToka, OktayTuretkan, “Accuracy of Contemporary Parametric Software Estimation Models”, 2013 IEEE.

[5] HimaniRastogi, MishaKakkar,“A Survey on Software Effort Estimation Techniques”,2014 IEEE.

[6] PoonamPandey, “Analysis of the Techniques for Software Cost Estimation”, 2013 IEEE

[7] TripoMatijevic,Ivana Ognjanovic, Ramo Sendelj, “Enhancement of Software Projects’ Function Point Analysis Based on Conditional Non-Functional Judgment”, Mediterranean Conference on Embedded Computing, MECO-2012.

[8] Chetan Nagar and Anurag Dixit, “Efforts estimation by combining the use case point and COCOMO”, IJCA journal, 2012

[9] Adanma C. Eberendu, “Software Project Cost Estimation: Issues, Problems and Possible Solutions”, International Journal of Engineering Science Invention,June, 2014

[10] Mohammad Azeeh, Software Cost Estimation Based on Use Case Points for Global Software Development”, CSIT, ©2013 IEEE

[11] S.chulani, B. Boehm, and B. Steece, “Bayesian Analysis of Emperical Software Engineering Cost Models,’ IEEE Trans. Software Eng., vol.25, no. 4, pp.573 583, 1999.

[12] Inderpreetkaurwalia and Manpreetkaur, “To Enhance the Effort Estimation Accuracy of COCOMO Model using Function Point, IJESRT, December 2013.

[13] Hill Peter (ISBSG) - Estimation Workbook 2 - published by International Software Benchmarking Standards Group ISBSG - Estimation and Benchmarking Resource Centre

[14] Briand, L. C. and I. Wieczorek (2002), ”Resource estimation in software engineering encyclopedia of software engineering”,J.J. Marcinak. New York, John Wiley & Sons: 1160-1196.

[15] Jørgensen, M. Shepperd, M. "A Systematic Review of Software Development Cost Estimation Studies"

[16] Abran, Alain; Moore, James W.; Bourque, Pierre; Dupuis, Robert; Tripp, Leonard L. (2004). Guide to the Software Engineering Body of Knowledge. IEEE.ISBN 0-7695-2330-7.

[17] Juan J.CuardradoGallego, Daniel Rodiguez, Software Project Effort Estimation Based on Multiple Parametric Models Generated Through Data Clustering,Journal of computer science& technology, 2007.

[18] Putnam, Lawrence H. (1978). "A General Empirical Solution to the Macro Software Sizing and Estimating Problem" IEEE transactions on software engineering, vol. se-4, NO. 4, pp 345-361.

[19] "Focus on Lawrence Putnam: A CAI State of the Practice Interview". Computer Aid, Inc. September 2006.

[20] Wei Lin Du, Danny Ho,Luiz Fernando Capretz “Improving Software Effort Estimation UsingNeuro-Fuzzy Model with SEER-SEM”, Global Journal of Computer Science and TechnologyVol. 10 Issue 12 (Ver. 1.0) October 2010

[21] JuhaKoskenkylä, “Cost estimation in global software development – review of estimation technique”,Department of Information and Service Economy, Aalto university,2012.

[22] Bente Anda ,Hege Dreiem, Dag I.K. Sjøberg and Magne Jørgensen “Estimating Software Development Effort based on Use Cases - Experiences from Industry”, Proc. Of 4th International Conference on The Unified Modeling Language, Modeling Languages, Concepts and Tools, pp. 487-502,2002.

[23] Smith, J. (1999). The estimation of effort based on use cases. Rational Software white paper.

[24] Ömer Faruk SARAÇ, Nevcihan DURU (2013) “A Novel Method For Software Effort Estimation: Estimating With Boundaries”, Proc. Of Institute of electrical and electronics engineers,pp. 1-5.