_____

Design & Analysis of Cost Estimation for New Mobile-COCOMO Tool for Mobile Application

Bhawana Verma

M.tech Scholer,

Depatment of Computer Science & Engineering

Kautilya Institute of Technology & Engineering, Jaipur

Email:bhawanav2110@gmail.com

Mo: 8387901800

Satish Kumar Alaria

Asst. Professor,

Depatment of Computer Science & Engineering

Kautilya Institute of Technology & Engineering, Jaipur

Email:satish.alaria@gmail.com

Mo: 9799611032

ABSTRACT: Software cost estimation is a resource forecasting method, which is required by the software development process. However, estimating the workload, schedule and cost of a software project is a complex task because it involves predicting the future using historical project data and extrapolating to see future values. For cost estimates for software projects, several methods are used. Among the various software cost estimation methods available, the most commonly used technology is the COCOMO method. Similarly, to calculate software costs, there are several cost estimating tools available for software developers to use. But these released cost estimation tools can only provide parameters (i.e. cost, development time, average personnel) for large software with multiple lines of code. However, if a software developer wants to estimate the cost of a small project that is usually a mobile application, the available tools will not give the right results. Therefore, to calculate the cost of the mobile application, the available cost estimation method COCOMO II is improved to a new model called New Mobile COCOMO Tool. The New Mobile COCOMO tool developed specifically for mobile applications is a boon for software developers working in small software applications because it only includes important multipliers that play a vital role in estimating the cost of developing mobile applications.

Therefore, the objective of this paper is to propose a cost estimation model with a special case of COCOMO II, especially for mobile applications, which calculates the person-month, the programmed time and the average personnel involved in the development of any mobile app.

_____*******_____

I. INTRODUCTION

A computer software or software can be defined as a combination of

- Data structures that allow programs to correctly manipulate information.
- Documentation that illustrates the action and use of the program.
- The instructions that in the execution provide preferred characteristics, functions and performance.

1.1 Time and Purpose of Software Cost Estimation

In the early stages of the project, an approximate estimate was made to help managers decide whether to make or purchase software and perform cost / use or balance analysis. In this estimate, the total cost and the schedule are significant.In the development process, estimating software costs and measuring together provide a tool for the project manager to monitor the progress of each phase of the software. These estimates require more details to be effective.

1.2 Potential Problems of Estimation

The project specification builds the foundation for all estimation work. Changes in requirements will result in changes in specifications. This is a very serious problem for both the estimator and the developer.In most projects, they are really only a small part of the entire code. Another important point of this approach is that other components are underestimated, such as the graphical user interface. All generated documents, in addition to the code, represent a large part of the overall workload.The starting point for software estimation [17] is the size of the project, be it a physical code line, a logical source code declaration, a function point or, sometimes, the three indicators. Once the project size is determined, it can be estimated according to the specific properties of the project in question.Software cost estimation [1] is a method to predict the resources needed for the software development process. To create accurate software cost estimates, knowledge of the following parameters must be processed.

1. The possible number of errors or defects that have the possibility of finding.
2. The speed at which requirements may change during development.
3. The sizes of the main deliverables, such as specifications, source code and manuals.
4. The capacity or capacity of the development team.

5. General expenses and salaries related to the development team.

Estimated cost types:

The estimated cost [14] can be classified into two groups: conceptual estimates and detailed estimates. Both of the above can be defined as follows:

1. Conceptual estimate: the conceptual estimate is also called parametric estimation. In this process, the graphic representation of an installation is developed after establishing the cost of the project.

2. Detailed estimate: the detailed estimate can be defined as a product of a process that aims to calculate the cost of a proposed construction project. To plan the estimate, the elements of the work are broken down in an orderly and logical manner and then the cost of each element is determined, which ends with the summary of the total.

The cost estimation procedure includes six steps. They are the following:

Step 1: Define the cost to estimate.

Step 2: The next step is to determine the cost factors. This is usually the most important step.

Step 3: Then consistent and accurate data is collected.

Step 4: Then the collected data is graphed.

Step 5: This step includes selecting the appropriate estimation method and then using it.

Step 6: And finally the accuracy of the estimated cost is evaluated.

If an appropriate cost estimation technique is executed, it definitely contributes to the accuracy of cost estimates.

1.3 Algorithmic Methods

Algorithmic methods are based on mathematical models that produce the estimated cost based on a series of variables, which are considered the main cost factors. Any algorithmic model has the form:

$$\text{Effort} = f(x_1, x_2, ..., x_n)$$

Where $\{x_1, x_2, ..., x_n\}$ denotes the cost factors. The existing algorithmic methods differ in two aspects: the selection of cost factors and the form of the function f.

COCOMO Models (Constructive Cost Model)

This family of models was proposed by Boehm. The models [2] have been widely accepted in practice. In COCOMO, the [20] size S code is given in thousands of LOC (KLOC) and the effort is in person-month.

1.**Basic COCOMO** : in this model based on software complexity, three sets of {a, b} are used.

- For simple, well-understood applications, the values are a = 2.4, b = 1.05
- For more complex systems, a = 3.0, b = 1.15
- For embedded systems, a = 3.6, b = 1.20

COCOMO Intermediate And Detailed COCOMO : In the intermediate COCOMO, an estimate of the nominal effort is obtained by using the power function with three sets of {a, b} with coefficients 'a' that are slightly different from those of the basic COCOMO:

- For simple applications well understood, a = 3.2, b = 1.05
- For more complex systems, a = 3.0, b = 1.15
- For embedded systems, a = 2.8, b = 1.20

COCOMO II : Perhaps the most significant difference with respect to the first COCOMO models is that the b exponent changes according to the following cost factors: development flexibility, team cohesion, process maturity, precedence and architecture or risk resolution [9] Other differences include recently added cost factors and models to solidify the software architecture and reduce risk.

The same COCOMO II, does not differentiate web applications and traditional applications, there are values of the parameters that will differ. In COCOMO II [5], the amount of effort in person-months, PM, is estimated by the formula:

$$PM = A \times Size^E \times \prod_{i=1}^{n} EM_i$$

$$\text{Where, } E = B + 0.01 \times \sum_{i=1}^{5} SF_i$$

The size of the application must be scaled according to the following five scale factors:

- Precedence (PREC)
- Flexibility of development (FLEX)
- Architecture / Risk Resolution (RESL)
- Team cohesion (TEAM)
- Maturity of the process (PMAT)

Cost factors are the characteristics of software development that affect the execution of a project. Unlike scale factors, cost factors are chosen based on their fundamental principles of linear impact on effort. Six effort multipliers were used in the COCOMO II model to standardize the development work.

- Reliability of the Required Software (RELY)
- Size of the database (DATA)
- Developed for reuse (RUSE)
- Documentation matching life cycle needs (DOCU)

28

- Runtime restriction (TIME)
- Master storage restriction (STOR)
- Volatility of the platform (PVOL)
- Analyst capacity (ACAP)
- Programmer's capacity (PCAP)
- Continuity of Personnel (PCON)
- Experience in Applications (APEX)
- Platform experience (PLEX)
- Language and experience in tools (LTEX)
- Use of software tools (TOOL)
- Multi-site development (SITE)
- Required development program (SCED)

1.4 Nominal - Time Estimation Equations

Both later design and early architecture models use the same functional form to estimate the amount of work and the calendar time required to develop a software project [4]. These nominal schedule formulas (NS) include cost factors.

$$PM_{NS} = A \times Size^E \times \prod_{i=1}^{n} EM$$

$$Where, E = B + 0.01 \times \sum_{j=1}^{5} SF_j$$

For the required development program, the amount of effort in person-months, PM_{NS} is estimated using the formula above.

The amount of time of the calendar, $TDEV_{NS,}$ that will take to develop the product is estimated by means of the formula:

$$TDEV_{NS} = C \times (PM_{NS})^F$$

$$Where, F = D + 0.2 \times 0.01 \times \sum_{j=1}^{5} SF_j$$

$$= D + 0.2 \times (E-B)$$

The value of n is 16 for the stress multipliers of the post-architecture model, Em_i, and 6 for the early design model, the number of SF_j represents exponential scale factors.

The values of A, B, C, D, $SF_{1,}$ and SF_5 for the Early Design model they are the same as for the Post-Architecture model. The EM values EM_1 ,........, EM_{16} for the Early Design model they are obtained by combining the values of their 16 Post-Architecture counterparts.

The value of A, B, C and D in the COCOMO II are:
A = 2.94, B = 0.91, C = 3.67, D = 0.28

The effort applied is measured by the number of person per month, which is also useful for estimating the cost of the project.

Cost = person-month × average work rate

1.5 Project Cost

It is the responsibility of software project managers [15] to control project budgets, so they should be able to estimate how much software development will cost. The main components of the project costs include:

- Effort costs
- Travel and training expenses.
- Hardware costs

Among the components of project costs, labor costs are the most difficult to estimate and control the administration costs and have the most significant impact on total costs. The cost of the software must be done objectively to accurately predict the cost of the contractor developing the software.

II. LITERATURE SURVEY

Software Cost Estimating (SCE) is a process of predicting the funds, schedule, workload and cost of any software system. Software cost estimation is an effective but critical process in software development and project development. The estimated cost of the project includes three types of costs, namely, labor costs, travel and training costs, and hardware costs. In these three cost efforts, cost is the most important cost. To calculate this cost, several estimation techniques can be used. Among the various methods available, the cost estimation method COCOMO II is the most accepted method. Some cost estimation tools have also been developed using COCOMO II. But all the research work is done for large projects.That is, no tool of this type is available for mobile applications.

In Mohammed MugahedAl_Qmase, M. In the RizwanJameel Qureshi document [26], the main focus is on the constructive cost model (COCOMO). Briefly presents the sub-models of COCOMO-COCOMO I and COCOMO II. Through this article, the author analyzes some case studies to evaluate the accuracy of the COCOMO I and COCOMO II models.

Jyoti G. Borade, Vikas R. Khalkar [27] discussed different aspects of the work of existing software projects and methods of estimating costs in their articles. In addition to cost estimates and workload, it also focuses on software metrics for cost estimates of software projects. According to the author, no available model can calculate the cost of software with high precision. In this article, the author analyzes the estimate of the test workload. This concept of estimating the test workload is a key part of the estimation

process, since it forms an important part of the overall software development effort.

In this article, Jyoti Mahajan, Simmi Dutta [28] mentioned that the concept of estimation accuracy has been discussed in several studies with the help of several formal estimation models. It focuses on the calculation of accurate calculations with the reuse of software as the main focus. Previously, formal estimation models were developed to measure lines of code and function points, but most did not improve the accuracy of estimates. In this article, the author describes the concept of reuse in software development using the concept of artificial neural network for estimating the workload. The authors propose a new model called COREAN for an efficient estimation of the workload with the improved RPROP algorithm and simulated annealing optimization technology, the accuracy of the model is further improved. Finally, comparing the proposed COREAN with the COCOMO II, it is concluded that the COREAN model is better than the COCOMO II. The main objective of this document is to calculate the exact amount of work by reusing the software.

III. PROBLEM & PROPOSED SOLUTION

3.1 Problem

The process of estimating software costs plays a vital role in the software industry. To develop applications, several cost estimation models are being used. The cost of the software includes three types of costs: project costs, hardware and software costs, and travel costs. Among these cost factors, the cost of the project is the main cost. The project costs are calculated per person.

Several cost estimation models can be used to help calculate the person's month, planning time, cost, which in turn helps estimate software costs. Some of the most commonly used software cost estimation methods are COCOMO, expert judgment and analogy methods. Of these various technologies, the most common method is COCOMO.

Based on the cost estimation techniques of COCOMO, a variety of cost estimating tools can be used. The user must enter the information required in the tool and the tool will automatically calculate the workload, the scheduled time and the profile. But these tools only correctly calculate these parameters for large applications. That is, those applications that have a lot of code. However, to calculate the month per person of a mobile application that generally has a small number of lines of code, there is no such model or tool available. The existing cost estimate model also does not generate an average staffing map for small projects. Then, the problem is to create a tool that can calculate the personal workload, the time of the calendar and the average staff, and the average of personnel for the mobile applications.

3.2 Proposed Solution

There are some problems in the COCOMO model of existing cost estimation. Because they include a large number of scaling factors and effort multipliers, several factors have no correlation in the context of mobile application cost estimates. Because these mobile applications are small projects with fewer lines of code.

If these models are used to calculate the cost of developing a mobile application, then it will take a long time. Therefore, some unrelated factors have been eliminated in the proposed cost estimation model for mobile applications, which makes the model efficient and accurate.

In addition, the suggested solution can generate a profile of the people profile of the mobile application, which can not be calculated with the available tools.

IV. RESULT ANALYSIS

Values for three different projects are implemented on the COCOMO II cost estimation tool as well as New Mobile COCOMO tool. The inputted values and the results for both the tools are shown below.

4.1 Input and Output Values of COCOMO II and New Mobile COCOMO Tool

The tables shown below are describing the values of SLOC for three different projects which are inputted in the corresponding two tools, along with the output values in the form of Effort (Person Month), Calendar Time (Months) and Average staffing.

TABLE 4.1
Input & Output values by COCOMO II Tool

Name of Project	SLOC	Values Calculated from COCOMO II Tool		
		Effort (Person Month)	Calendar Time (Months)	Average Staffing
Weather	2500	4.3197	5.9	Cannot generate a staffing profile due to small project
Stock Exchange	3100	3.5330	5.56	Cannot generate a staffing profile due to small project
Call Manager	2120	2.02	4.6	Cannot generate a staffing profile due to small project

30

TABLE 4.2

Input & Output values by New Mobile COCOMO Tool

Name of Project	KSLOC	Values Calculated from New Mobile COCOMO Tool		
		Effort (Person Month)	Calendar Time (Months)	Average Staffing
Just Weather	2.5	4.3197	5.9305	Generates Avg. Staff. Chart
Stock Exchange	3.1	3.5330	5.5521	Generates Avg. Staff. Chart
Call Manager	2.12	2.0283	4.6281	Generates Avg. Staff. Chart

4.2 Proposed Formula for New Mobile COCOMO Tool

The below proposed COCOMO II model formula calculate the effort month for mobile applications:

$$PM = A * Size^E * \prod_{i=1}^{11} EM_i$$

where,

$$E = B + 0.01 * \sum_{j=1}^{2} SF_j$$

4.3 Snapshots showing comparison between COCOMO II and New Mobile COCOMO Tool

The snapshots shown below are describing the individual implementation of scale and effort multipliers on COCOMO II and New Mobile COCOMO tool. The disadvantage of using COCOMO II cost estimation tool for mobile applications can be easily understood with the help of the snapshots for each individual project.

4.3.1 Comparison between COCOMO II and New Mobile COCOMO Tool for Project 1 (Weather)

In this, the ideal values of the selected multipliers are inputted in the COCOMO II tool. And similarly the same values of the selected multipliers are inputted in the New Mobile COCOMO tool. The ideal values are chosen according to the project undertaken i.e. just weather. Then based on the resultant values of the effort, schedule time and staffing profile from both the tools, the comparison is done which is explained below with the help of the snapshot in fig. 4.1 and 4.2 .
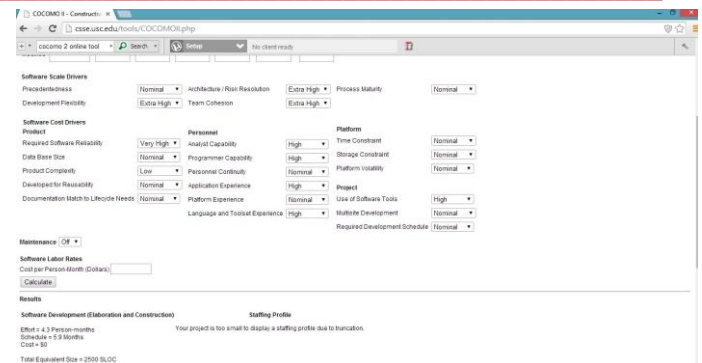

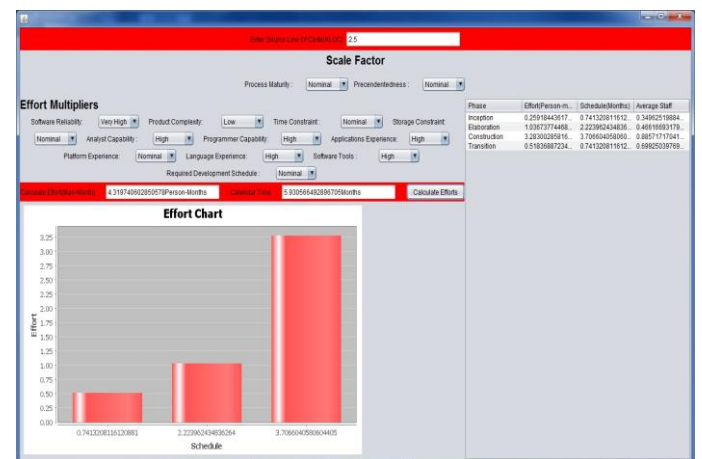Figure 4.1 Screenshot of COCOMO II Tool for Project1 (Weather)

Figure 4.2 Screenshot of the New Mobile COCOMO Tool for Project 1(Weather)

4.3.2 Comparison between COCOMO II and New Mobile COCOMO Tool for Project 2 (Stock Exchange)

In this, the ideal values of the selected multipliers are inputted in the COCOMO II tool. And similarly the same values of the selected multipliers are inputted in the New Mobile COCOMO tool. The ideal values are chosen according to the project undertaken i.e. stock exchange. Then based on the resultant values of the effort, schedule time and staffing profile from both the tools, the comparison is done which is explained below with the help of the snapshot in fig. 4.3 and 4.4
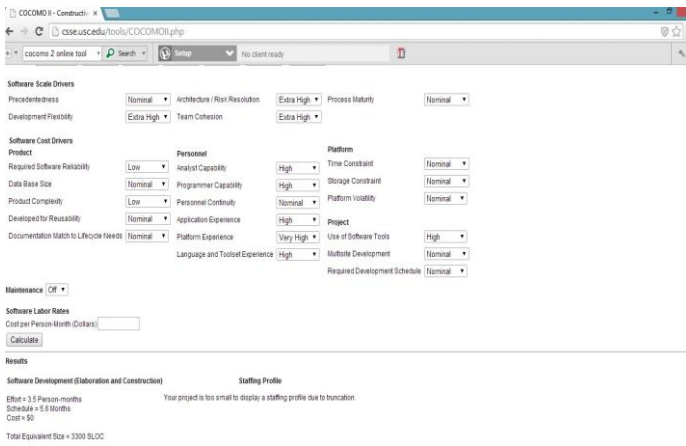
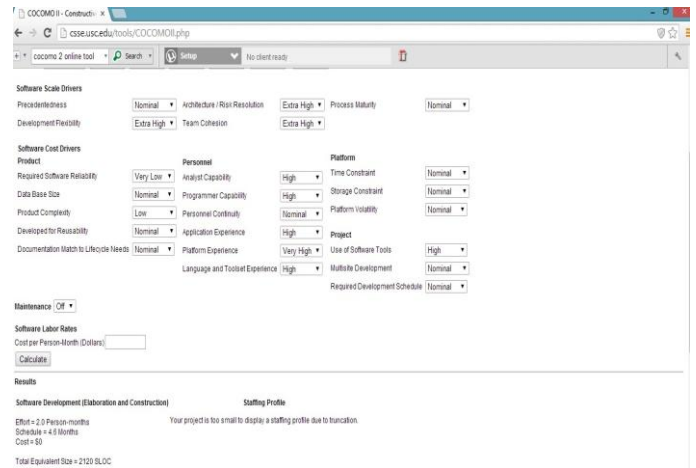Figure 4.3 Screenshot of COCOMO II Tool for Project 2 (Stock Exchange)

Figure 4.5 Screenshot of COCOMO II Tool for Project 3 (Call Manager)
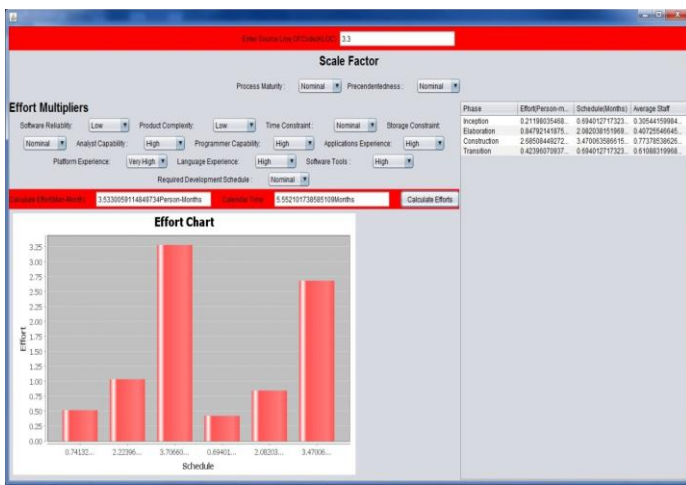

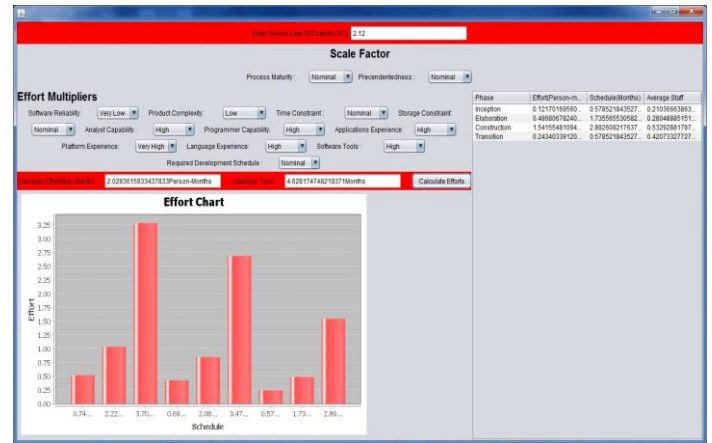Figure 4.4 Screenshot of New Mobile COCOMO Tool for Project 2 (Stock Exchange)

Figure 4.6 Screenshot of New Mobile COCOMO Tool for Project 3 (Call Manager)

4.3.3 Comparison between COCOMO II and New Mobile COCOMO Tool for Project 3 (Call Manager)

In this, the ideal values of the selected multipliers are inputted in the COCOMO II tool. And similarly the same values of the selected multipliers are inputted in the New Mobile COCOMO tool. The ideal values are chosen according to the project undertaken i.e. call manager. Then based on the resultant values of the effort, schedule time and staffing profile from both the tools, the comparison is done which is explained below with the help of the snapshot in fig. 4.5 and 4.6 .

4.4 COCOMO Derivative Models

The COCOMO family tree is shown in the diagram below. The Proposed COCOMO model i.e. New Mobile COCOMO is indicated in the diagram which also belongs to the COCOMO family.
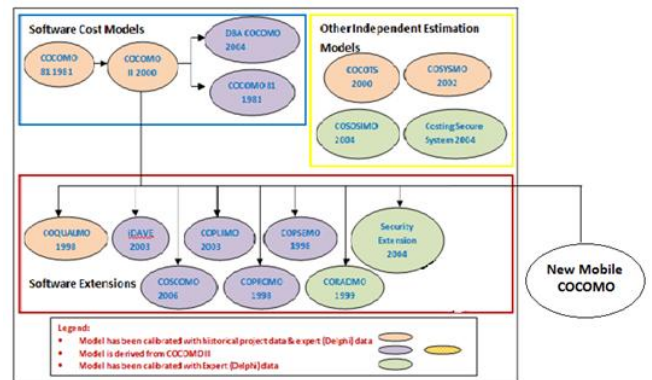

Figure 4.7 COCOMO Derivative Models

The above COCOMO family tree describes the various models derived from COCOMO model. Along with the new

32

proposed model for mobile applications i.e. New Mobile COCOMO.

V.   CONCLUSION

In this work, several software cost estimation methods were discussed and a detailed study was carried out on the cost estimation method of COCOMO II and its importance. From the study of the COCOMO II method it can be concluded that there are several open source tools available to estimate the cost of large applications, that is, those applications that have several lines of code. But for a software developer, who is involved in the development of mobile applications that have few lines of code, the use of available COCOMO II cost estimation tools available is useless. Since then, these tools provide the expected results only for those applications that have several lines of code. So, calculating parameters for mobile applications is a problem for the developer. Therefore, to overcome this problem, a new improved tool of the cost estimation tool COCOMO II called New Mobile COCOMO is prepared. This tool provides the developer with all the expected parameters required for cost estimation. Therefore, it can be concluded that estimating the cost of mobile applications is not a problem for developers now, since there is a tool available as New Mobile COCOMO.

REFERENCES

[1]. Anil Bhardwaj, Anita Sharma and TrilokiNathSharma,"Software cost estimation for web projects",*Int. J. Agricult. Stat. Sci.*, *Vol. 7*, *No. 1*, *pp. 281-287, 2011.*

[2]. TN Sharma, "Analysis of Software Cost Estimation using COCOMO II", International Journal of Scientific & Engineering Research Volume 2, Issue 6, June-2011

[3]. Dr. Pradeep Kumar, SapnaMathur "Comparative Analysis of Cost Estimation for Agent Oriented Software & Traditional Software", International Journal of Innovations in Engineering and Technology, Special Issue - ICAECE-2013, pp. 10 -18.

[4]. ChanderDiwaker, AsthaDhiman, "Size and Effort Estimation Techniques for Software Development", International Journal of Software and Web Sciences, Vol-4, pp. 36-40, March-May, 2013

[5]. Barry Boehm, Bradford Clark, Ellis Horowitz, Chris Westland, "Cost models for future software life cycle processes", vol-1, pp 57-94, 1995.

[6]. HartmutSteck Winter, "The four basic constituents of a semiparametric engineering effort estimation", The11th International Scientific Conference Trends in Business Management Systems - TBMS 2008, 9th– 11th December 2008.

[7]. T.N.Sharma, Anil Bhardwaj, Anita Sharma,"A Comparative study of COCOMO II and Putnam models of Software Cost Estimation", International Journal of Scientific & Engineering Research, Volume 2, Issue 11, November-2011.

[8]. Roger Smith, Lacey Edwards, "COCOMO-SCORM Interactive Courseware Project Cost Modeling", International Council of Systems Engineering Conference, 2006.

[9]. Ye Yang, Zhihao Chen, Ricardo Valerdi, Barry Boehm, "Effect of Schedule Compression on Project Effort", Center for Software Engineering, University of Southern California (USC-CSE) Los Angeles, CA 90089-0781, USA.

[10]. B. Boehm, B. Clark, E. Horowitz and C.Westland,"Cost models for future software life cycle processes: COCOMO 2.0", Springer Netherlands, Annals of Software Engineering, vol 1, pp. 57-94. 1995.

[11]. Basavaraj, M.J. and Shet, K.C. (2008). Empirical Validation of Software Development effort multiplier of Intermediate COCOMO Model. Journal of Software, vol.3, No. 5 pp. 65-71

[12]. T.N.Sharma, Anil Bhardwaj, G.R.Kherwa, "Statistical Analysis of various models of Software Cost Estimation",International Journal of Engineering Research and Applications, Vol. 2, Issue 3, pp.683-685, May-Jun 2012.

[13]. Demirors, Gencel, "Conceptual Association of Functional Size Measurement Methods", IEEE Society, Volume 26, Issue 3, p.p. 71 – 78, May-June 2009.

[14]. VahidKhatibi, "Software Cost Estimation Methods: A Review Journal of Emerging Trends in Computing and Information Sciences", Volume 2 No. 1, 2010-11.

[15]. Ghulam QadirMemon, S.M. Ahmed Bari, "Impact of Analyst and Programmer Capability on Software Development Cost", Journal of Information and Communication Technology, Vol. 1, no. 2, pp. 21-34, 2007.

[16]. Emilia Mendes, Nile Mosley, Steve Counsell, "Investigating Early Web Size Measures for Web Cost Estimation", Journal of Systems and Software, Vol. 77, Issue 2, pp. 157-172, 2005.

[17]. Chris F. Kemerer and Michael W. Patrick, "Software Engineering Productivity Handbook", pp. 175-190, 1993.

[18]. Frank Niessink and Hans van Vliet, " Two Case Studies in Measuring Software Maintenance Effort", Published in the proceedings of the International Conference on Software Maintenance, Bethesda, Maryland, USA, pp.76-85, November-16-20, 1998.

[19]. Boehm, B. &Papaccio, P. 1988. "Understanding and controlling software costs".IEEE Transactions on Software Engineering 14 (10), 1462-1477, 1988

[20]. Boehm, B., Horowitz, E., Madachy, R., Reifer, D., Clark, B., Steece, B., Brown,A.W., Chulani, S. &Abts, C, "Software Cost Estimation with COCOMO II". Prentice Hall, 502 p., 2000.

[21]. Abedallah Zaid, Mohd Hasan Selamat, Abdual Azim Abd Ghani, RodziahAtan and Tieng Wei Koh," Issues in Software Cost Estimation", International Journal of

Computer Science and Network Security, VOL.8 No.11, November 2008.

[22]. M.A. Al-Hajri, A.A.A. Ghani, M.S. Sulaiman, M.H. Selamat," Modification of standard function point complexity weights system," Journal of Systems and Software, Vol.74 ,195–206, (2005).

[23]. Wei Xiaa, Luiz Fernando Capretz," A new calibration for Function Point complexity weights,": Journal of Systems and Software, Vol. 50, pp.670- 683 ,2008.

[24]. J. Asundi. " The Need for Effort Estimation Models for Open Source,",ACMvol30,pp1-3,2005.

[25]. I Wieczorek, M Ruhe," How valuable is company specific data compared to multi-company data for software cost estimation? "METRICS.02 , IEEE, pp 237- 246, 2002.

[26]. Mohammed MugahedAl_Qmase, M. RizwanJameel Qureshi, "Evaluation of the Cost Estimation Models: Case Study of Task Manager Application", I.J.Modern Education and Computer Science, Vol-8, pp 1-7, 2013.

[27]. Jyoti G. Borade ,Vikas R. Khalkar, "Software Project Effort and Cost Estimation Techniques", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 8, August 2013.

[28]. Jyoti Mahajan, Simmi Dutta, "COREAN: A proposed Model for Predicting Effort Estimation having Reuse", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-6, January 2013.s

34