Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 6, June 2014, pg.547 – 557

RESEARCH ARTICLE

Comparative Analysis of Different Software Cost Estimation Methods

Tannu[1], Yogesh Kumar[2]

[1]Student M.Tech, Department of Computer Science and Engineering, U.I.E.T., M.D.U. Rohtak, India
[2]Assistant Professor, Department of Computer Science and Engineering, U.I.E.T., M.D.U. Rohtak, India

[1]Email:- tannuchugh27@gmail.com; [2]Email:- yogs_crsce@yahoo.com

Abstract- The importance of software cost estimation has been increasing gradually over last three decades. Software cost estimation is related to how long and how many people are required to complete a software project. Software cost estimation is the process of predicting the effort and cost required to develop a software system. The basic input for the software cost estimation is coding size and set of cost drivers, the output is Effort in terms of Person-Months (PM's) and cost. One of the problems in software cost estimation is how we can evaluate the cost estimation model. A key factor in selecting a cost estimation model is the accuracy of its estimates. Unfortunately, despite the large body of experience with estimation models, the accuracy of these models is not satisfactory. This paper provides a detail overview of existing software cost estimation models and techniques. This paper present the advantage and disadvantage of various cost estimation method. We also focus the importance of accurate cost estimation. Here, cost estimation tool Costar is used for estimation.

Keywords--- PM-person-months, costar tool, COCOMO, software cost estimation model, accuracy

1. INTRODUCTION

[1]The software cost estimation is the important concept of software development process. Software cost estimation means how many people are required to complete a software project within time and budget. Accurate cost estimation helps to complete the projects within time and within budget. The estimation process consists of size estimation, effort estimation, schedule estimation and at last estimate the overall cost. [2]In order to achieve the attributes (like budget, time and resources), there is a need to measure all the requirements. With consideration of all the requirements, the cost estimation process becomes easier and will produce accurate result. But unfortunately the development of software cost estimation process is difficult and not accurate. In order to manage budget and schedule of software projects [3], various software cost estimation models have been developed. Accurate software cost estimates are critical to both

developers and customers. They can be used for generating request for proposals, contract negotiations, scheduling, monitoring and control.

[4]Cost estimation is a process or an approximation of the probable cost of a product, program, or a project, computed on the basis of available information. Accurate cost estimation is very important for every kind of project, if we do not estimate the projects in a proper way; result the cost of the project is very high sometimes it will be reached 150-200% more than the original cost [5]. So in that case it is very necessary to estimate the project correctly. The Cost for a project is a function of many parameters. Size is a primary cost factor in most models and can be measured using lines of code (LOC) or thousands of delivered lines of code (KDLOC) or function points. A number of models have been evolved to establish the relation between size and effort for Software Cost Estimation.

Importance of cost estimation accuracy

Accurate cost estimation in the software industries is really important due to the following reasons [6] :

1. The needs of resources should be completely matched with the real needs for easily manage and control of the project.
2. Customer always expects that the actual development cost should be matched with the estimated software development cost.
3. The overall business plan of a software organization can be improved with accurate cost estimation, as it will lead to an efficient use of the resources.
4. Cost estimation process is used to determine which resources are required for the project and how to better utilize resources.
5. The accurate cost estimation process is necessary for defining the resources needed to produce, verify and validate the software products and for managing the software development activities.
6. It can be used to support re planning.

There are many reasons that make cost estimation process difficult.

1. It is very difficult to estimate the cost of software development. One of the first steps in any estimate is to understand and define the system to be estimated.
2. There are several interrelated factors which affect the cost estimation process in the software development like complexity of the project.
3. A software cost estimator cannot obtain reliable estimate in early stage because of lack of detailed information at early stage.
4. Lack of information about past projects of cost measurement that means historical data is sometimes incomplete, inconsistent, or inaccurate.
5. Lack of trained estimators and estimators with the necessary expertise.
6. The too low effort estimates may lead to project management problems, delayed deliveries, cost overruns and low software quality, too high effort estimates may lead to lost business opportunities and inefficient use of resources.

Cocomo81 and Cocomo II use costar tool. Costar tool is used to estimate the effort, duration and cost. It also helps us to compare the various methods of cost estimation. In the section 2 we will

describe about the estimation technique, section 3 include the advantages and disadvantages of existing methods and section 4 include how to evaluate the result of an estimation method and section 5 include conclusion.

2. ESTIMATION TECHNIQUES

[7]Generally there are many methods and techniques for software cost estimation which are divided into two groups:

1. Non Algorithmic
2. Algorithmic

In the Algorithmic method we use a formula for calculating the cost estimation. The formula is created by combining related cost factors in the various models. Non-algorithmic methods do not use any formula to calculate the software cost estimation. Both groups are useful for performing the accurate estimation. If the requirements are known better, their performance will be better. In this section, some popular estimation methods are discussed.

Non Algorithmic

1. *Expert Judgment:*

In the Expert judgment technique we can estimate by getting advice from different expert who have an extensive experiences in similar projects. This method is usually used when there is limitation in finding data and gathering the requirements. Consultation is the basic issue in this method. One of the most common methods which works according to this technique, is Delphi. Delphi consists of group of expert and coordinator and conduct an especial meeting among the project experts and tries to achieve the true information about the project from their debates. Delphi includes some steps:

i. The coordinator provide an estimation form to each expert.

ii. Each expert complete their individual estimate anonymously (without discussing with others)

iii. The coordinator gathers all forms and sums up them (including mean or median) on a form and asks experts to start iteration.

iv. Coordinator prepares and distributes a summary of the response of all the estimators in an iteration form.

v. Steps (ii-iii) are repeated until an approval is gained.

The wideband Delphi Technique has subsequently been used in a number of studies and cost estimation activities. It has been highly successful in combining the free discuss advantages of the group meeting technique

2. *Estimation By Analogy:*

In this method, several similar completed software projects are analyzed and estimation of effort and cost are done according to their actual cost and effort. Estimation by this technique is accomplished at the total system levels and subsystem levels. By assessing the results of previous actual projects, we can estimate the cost and effort of a similar project. The steps of this method are considered as:

i. Choosing of analogy.

ii. Investigating similarities and differences.

iii. Examining of analogy quality.

iv. Providing the estimation

3. Parkinson's Law:

Using Parkinson's principle, the cost is determined not estimated. The cost is determined by the available resources rather than based on an objective assessment. If the software project has five people and 12 months to deliver the project, the effort is estimated to be 60 person-months. This method sometimes gives good estimation, it is not recommended as it may provide very unrealistic estimates. Also, this method does not promote good software engineering practice

4. Price-to-win:

The software cost is estimated to be the best price to win the project. The estimation is based on the customer's budget rather than software functionality. For example, if a reasonable estimation for a project costs 100 person-months but the customer can only afford 60 person-months then estimator do the work in such a way to complete the project in 60 person months effort in order to win the project. This is again not a good practice and cause a bad delay of delivery or force the development team to work overtime.

Algorithm Method

Algorithmic method use formula and provide some mathematical equations to perform software estimation. These mathematical equations are based on historical data and use inputs such as Source Lines of Code (SLOC), number of functions to perform, and other cost drivers such as language, design methodology etc. The algorithmic method develops a lots of models like COCOMO models, Putnam model, and function points based models. There are a variety of different models available, the best known are Boehm's Albrecht's' function points and COCOMO model.

1. Function Point Size Estimates:

[8]At first, Albrecht (1983) presented Function Point metric to measure the functionality of project. In this method, estimation is done by determination of below indicators:

Number of user input, Number of user outputs, Number of logic files, Number of inquries, Number of Interfaces

A Complexity Degree which is between 1 and 3 is defined for each indicator. 1, 2 and 3 stand for simple, medium and complex degree respectively. Also, it is necessary to define a weight for each indicator which can be between 3 and 15.

At first, the number of each mentioned indicator should be tallied and then complexity degree and weight are multiplied by each other. Generally, the unadjusted function point count is defined as below:

$$\mathrm{UFC}=\sum_{i=1}^{5}\sum_{j=1}^{3}N_{ij}W_{ij}$$

where Nij is the number of indicator i with complexity j and; Wij is the weight of indicator i with complexity j. According to the previous experiences, function point could be useful for software estimations because it could be computed based on requirement specification in the early stages

of project. To compute the FP, UFC should be multiplied by a Technical Complexity Factor (TCF). Each component can change from 0 to 5. 0 and 5 indicate that the component has no effect on the project and the component has strong effect and very important respectively. Finally, the TCF is calculated as:

$$TCF = 0.65+0.01(SUM (Fi))$$

The range of TCF is between 0.65 (if all Fi are 0) and 1.35 (if all Fi are 5). Ultimately, Function Point is computed as

$$FP=UFC*TCF$$

2. Cocomo Model:
Model1(Basic COCOMO Model):

[9]The basic COCOMO model is a static model that computes software development effort (and cost) as a function of program size expressed in estimated lines of code(LOC).This model has three development modes that is organic, semidetached and embedded. To determine the initial effort in person-months the equation used is of the type

$$EFFORT = x* (KLOC)y$$

$$TDEV=2.5*(EFFORT)z$$

TDEV is the estimated time to develop the software expressed in month. The value of constants x, y and z depend on the project type. It has following three classes of software projects describe in table I[10].

TABLE I

Basic COCOMO	**x**	**y**	**z**
Organic	2.4	1.05	0.38
Semi-detached	3.0	1.12	0.35
Embedded	3.6	1.20	0.32

Model2(Intermediate COCOMO Model):

Intermediate COCOMO Model computes software development effort as a function of program size and set of cost drivers that include subjective assessment of the products, computer, personnel and project attributes.The same basis equation for the intermediate cocomo model is used, but it include 15 cost drivers are rated on a scale of very low to very high that is used to calculate the effort multiplier and each of them returns an adjustment factor which multiplied yields in the total EAF (Effort Adjustment Factor). The adjustment factor is 1 is consider as normal. The intermediate COCOMO model takes the form:

$$EFFORT = x* (KLOC)y * EAF$$
$$TDEV=2.5(EFFORT)z$$

In addition to the EAF, the model parameter "x" is slightly different in Intermediate COCOMO from the basic model. The parameter "y" remains the same in both models

TABLE II

Intermediate COCOMO	x	y	z
Organic	3.2	1.05	0.38
Semi-detached	3.0	1.12	0.35
Embedded	2.8	1.20	0.32

Model 3 (Detailed COCOMO Model):

The detailed COCOMO Model incorporates all characteristics of the intermediate version with an assessment of the cost driver‘s impact on each step (analysis, design, etc) of the software engineering process.

Cocomo II Model:

[11]COCOMO-II is the latest version of COCOMO that predicts the amount of effort based on Person-Month (PM) in the software projects. It uses function point or line of code as the size metrics, and composes of 17 Effort Multipliers (shown in Table III) and 5 scale factors (shown in Table IV)

Some rating levels are defined for scale factors including very low, low, nominal, high, very high and extra high. A quantitative value is assigned to each rating level as its weight.
COCOMO II has some special features, which distinguish it from other ones. The Usage of this method is very wide and its results usually are accurate.

TABLE III Cocomo II Cost Driver

Attributes	Type	Very low	Low	Nominal	High	Very High	Extra High
RELY	Product	0.82	0.92	1.00	1.10	1.26	-
CPLX	Product	0.73	0.87	1.00	1.17	1.34	1.74
DOCU	Product	0.81	0.91	1.00	1.11	1.23	-
DATA	Product	-	0.90	1.00	1.14	1.28	-
RUSE	Product	-	0.95	1.00	1.07	1.15	1.24
TIME	Compute r	-	-	1.00	1.11	1.29	1.63
PVOL	Compute r	-	0.87	1.00	1.15	1.30	-
STOR	Compute r	-	-	1.00	1.05	1.17	1.46
ACAP	Personne l	1.42	1.19	1.00	0.85	0.71	-
PCON	Personne l	1.29	1.12	1.00	0.90	0.81	-
PCAP	Personne l	1.34	1.15	1.00	0.88	0.76	-
PLEX	Personne l	1.19	1.09	1.00	0.91	0.85	-

APEX	Personnel	1.22	1.10	1.00	0.88	0.81	-
LTEX	Personnel	1.20	1.09	1.00	0.91	0.84	-
TOOL	Project	1.17	1.09	1.00	0.90	0.78	-
SCED	Project	1.43	1.14	1.00	1.00	1.00	-
SITE	Project	1.22	1.09	1.00	0.93	0.86	0.80

(-)Value is not described

TABLE IV SCALE FACTORS

Precedentedness (PREC)	Reflects the previous experience of the organization
Development Flexibility (FLEX)	Reflects the degree of flexibility in the development process.
Risk Resolution (RESL)	Reflects the extent of risk analysis carried out.
Team Cohesion (TEAM)	Reflects how well the development team knows each other and work together.
Process maturity (PMAT)	Reflects the process maturity of the organization.

3. ADVANTAGE AND DISADVANTAGE OF EXISTING METHODS

[12]Here, we describe the advantages and disadvantages of existing cost estimation methods. This description could be useful for choosing an appropriate method in a particular project. Table V Shows a comparison of mentioned methods for estimation. For doing comparison, the popular existing estimation methods have been selected.

TABLE V

Method	**Type**	**Advantages**	**Disadvantages**
COCOMO II	Algorithmic	It provides more support for modern software development processes and an updated project database. Provide support to mainframe, code reusability and batch processing.	It cannot estimate the effort at all the different phase of SDLC. Its prediction is .68 which is quite good.
COCOMO	Algorithmic	Clear results, very common	Much data is required, It is not suitable for large project,
Expert Judgment	Non-Algorithmic	Fast prediction, Adapt to especial projects	Its success depend on expert, Usually is done incomplete
Analogy	Non-Algorithmic	Works based on actual experiences, having especial expert is not important	A lots of information about past projects is required, In some situations there are no similar project
Parkinson	Non-Algorithmic	Correlates with some experience	Reinforces poor practice
Price to win	Non-Algorithmic	Often gets the contract	Generally produces large overruns

4. HOW TO EVALUATE THE RESULT OF AN ESTIMATION METHOD

After knowing estimation methods and comparing them with each other, illustrating their abilities via some actual projects seems to be useful. The acceptance of using these methods has been a challenge since many years ago. In this section, we try to show the minimum distance between estimated parameters and actual ones in an experience.

Two equations are used to estimate effort and schedule as below:

$$EFFORT=2.94*EAF*(KSLOC)^{E}$$

Where

EAF Is the Effort Adjustment Factor derived from the Cost Drivers
E Is an exponent derived from the five Scale Drivers

Effort Adjustment Factor

The Effort Adjustment Factor in the effort equation is simply the product of the effort multipliers corresponding to each of the cost drivers for your project.

COCOMO II Schedule Equation

The COCOMO II schedule equation predicts the number of months required to complete your software project. The duration of a project is based on the effort predicted by the effort equation:

$$Duration = 3.67 * (Effort)^{SE}$$

Where

Effort Is the effort from the COCOMO II effort equation
SE Is the schedule equation exponent derived from the five Scale Drivers

An actual estimation by Cocomo II

Since COCOMO II is the most popular method used for estimation, in this section, a real project cost estimation is demonstrated based on COCOMO II metrics. Table shows the cost drivers and their adjusted amounts which are related to a real project. The scope of activities in mentioned organizations is banking, insurance, web development, communication and so on…

TABLE VI Effort Multiplier

Attributes Cost Driver	Degree	Value
RELY	Nominal	1.00
CPLX	Nominal	1.00
DOCU	Nominal	1.00
DATA	High	1.14
RUSE	Nominal	1.00
TIME	Nominal	1.00
PVOL	Nominal	1.00

STOR	Nominal	1.00
ACAP	Low	1.42
PCON	Nominal	1.00
PCAP	High	0.88
PLEX	Low	1.09
APEX	Nominal	1.00
LTEX	Low	1.09
TOOL	Nominal	1.00
SCED	Very High	1.00
SITE	Nominal	1.00

Table VII shows the scale factors values.

TABLE VII

Precedentedness (PREC)	Generally Familiar
Development Flexibility (FLEX)	Some Relaxation
Risk Resolution (RESL)	Little (20%)
Team Cohesion (TEAM)	Basically cooperative
Process maturity (PMAT)	SEI CMM Level1(upper half)

TABLE VIII ESTIMATED EFFORT AND COST OF DIFFERENT TYPES OF MODELS

Pr. no	**Size**	**Cocomo II.2000_ Waterfall Effort**	**Cocomo II.2000_ Waterfall Cost**	**Cocomo II.2000_ MBASE Effort**	**Cocomo II.2000_ MBASE Cost**	**Early Design.2000_ Waterfall Effort**	**Early Design.2000 _Waterfall Cost**	**REVIC_9.2 Effort**	**REVIC_9.2 Cost**	**Cocomo 87 Effort**	**Cocomo 87 cost**
1	1028	4.6	23.0	5.1	22.8	3.2	16.2	6.7	40.0	3.7	18.3
2	414	1.7	8.5	1.9	8.4	1.2	6.0	2.4	14.5	1.3	6.6
3	345	1.4	6.9	1.5	6.9	1.0	4.9	2.0	11.8	1.1	5.4
4	156	0.6	2.9	0.6	2.9	0.4	2.0	0.8	4.8	0.4	2.2
5	113	0.4	2.0	0.4	2.0	0.3	1.4	0.6	3.4	0.3	1.5

In this table we are comparing the different models. As seen in the table, as the size decreased, effort and cost are also decreased. The real cost of this estimation is 5000 taken as an average monthly developer salary. The report says that project cost is $23,000 and project duration is 4.6 months (in cocomo II .2000_Waterfall model) which is very near to actual estimation(23000 cost for 4.6 month means 5000 per month).

Accuracy depends on amount of reliable information. To produce better estimate, we must improve our understanding of project attribute and their casual relationship.

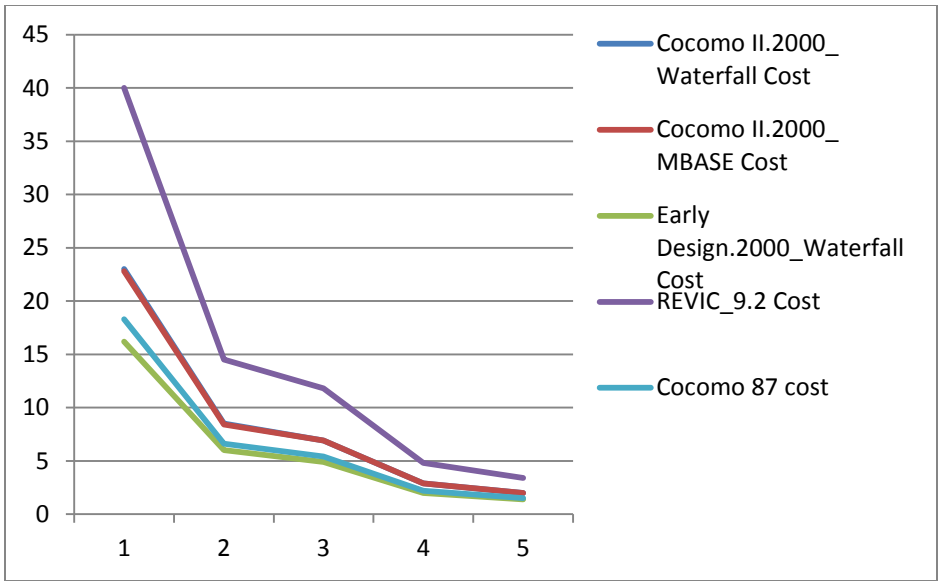

Fig1: Estimated Cost

The horizontal line show duration and vertical line show cost in thousand. The model that gives best value is cocomo II.2000_ waterfall model. The value of cocomo II.2000_ waterfall model and cocomo II.2000_MBASE model are nearly same sothat the cocomo II.2000_waterfall model cannot be seen in the graph.The original cost is 5000 per month.

5. CONCLUSION

This paper provides a comprehensive overview of different types of software cost estimation methods and also describes the advantages and disadvantages of these methods. This paper also presents some of the relevant reasons that cause inaccurate estimation. To produce a meaningful and reliable estimate, we must improve our understanding of software project attributes and their causal relationships, develop effective ways of measuring software complexity and the cost estimation process needs to be thoroughly arranged and carefully followed. This paper provide the use of cocomo II for estimation of software project effort and cost. We have use costar tool for estimation because it can use different different methods that can be help us to classify the data easily. We compare the result of different methods. It is observed from the result that COCOMOII.2000_waterfall model give better result. There is no estimation method which can produce the best estimates in all various situations and each technique can be suitable in the special project. It is necessary understanding the principals of each estimation method to choose the best. The search for reliable, accurate and low cost estimation methods must continue. Trying to improve the performance of the existing methods and introducing the new methods for estimation based on today's software project requirements can be the future works in this area.

REFERENCES

[1]Sweta Kumari , Shashank Pushkar," Performance Analysis of the Software Cost Estimation Methods", International Journal of Advanced Research in Computer Science and Software Engineering Volume 3, Issue 7,July 2013

[2] Zulkefli Mansor , Saadiah Yahya, Noor Habibah Hj Arshad, "Review on Traditional and Agile Cost Estimation Success Factor in Software Development Project", International Journal on New Computer Architectures and Their Applications (IJNCAA) 1(3): 942-952 2011 (ISSN: 2220-9085)

[3]L.H. Putnam,"A general empirical solution to the macro software sizing and estimation problem", IEEE Transactions on Software Engineering, pp. 345–361, July 1978.

[4]Sweta Kumari, Shashank Pushkar,"Comparison and Analysis of Different Software Cost Estimation Methods", *(IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 4, No.1, 2013*

[5]Narendra Sharma and Ratnesh Litoriya, "Incorporating Data Mining Techniques on Software Cost Estimation: Validation and Improvement, " International Journal of Emerging Technology and Advanced Engineering, 2012, vol.-2

[6]A.L. Lederer, and J. Prasad," Causes of Inaccurate Software Development Cost Estimates", Journal of Systems and Software, pp. 125-134, 1995

[7]Josephin.M.Baby Anusha M.E , Rajeswari Mukesh ," A Study On Software Cost Estimation", international journal of emerging trends and Technology in computer science(IJETTCS) special issue ISSN 2278-6856,2013.

[8]Vahid Khatibi, Dayang N. A. Jawawi," Software Cost Estimation Method:A Review", *Journal of emerging trends in computing and information sciences*, Volume 2 No. 1.

[9]Magne J and Martin S,"A Systematic Review of Software Development Cost Estimation Studies", IEEE Transactions On Software Engineering, Vol. 33, No. 1, pp. 33-53, January 2007.

[10]M.V. Deshpande and S.G. Bhirud,"Analysis of Combining Software Estimation Techniques", International Journal of Computer Applications (0975 – 8887) Volume 5 No.3, 2010

[11] T.N.Sharma,''Analysis of software cost estimation using COCOMO II", International Journal of Scientific & Engineering Research Volume 2, Issue 6, June-2011

[12] Omprakash Tailor[1], Jyoti Saini[2], Mrs. Poonam Rijwani[3] ," Comparative Analysis Of Software Cost And Effort Estimation Methods: A Review", International Journal of Computer Science and Mobile Computing,Vol. 3Issue. 4, April- 2014, pg. 1364-1374.

