International Journal of Scientific Research and Modern Education (IJSRME)
Impact Factor: 6.225, ISSN (Online): 2455 – 5630
(www.rdmodernresearch.com) Volume 2, Issue 1, 2017

A RELATIVE STUDY ON COST ESTIMATION TECHNIQUES

K. Jayapratha* & M. Muthamizharasan**
* Assistant Professor, Department of Computer Science, Dharmapuram Adhinam Arts College, Mayiladuthurai, Tamilnadu
** Associate Professor, Department of Computer Science, A.V.C College (Autonomous), Mayiladuthurai, Tamilnadu

Cite This Article: K. Jayapratha & M. Muthamizharasan, "A Relative Study on Cost Estimation Techniques", International Journal of Scientific Research and Modern Education, Volume 2, Issue 1, Page Number 183-186, 2017.

Abstract:

Software Cost Estimation is one of the most important part in software development. It involves in estimating the effort and cost in terms of money to complete the software development. Software Cost Estimation is very important when lines of code for the particular project exceeds certain limit, also when the software deployed with too many bugs and uncovered requirements the project will go incomplete. Software cost estimation of a project plays a vital role in acceptance or rejection of its development. There have been lot of Software cost estimation models in use. Appropriate cost estimation guarantees software to finish on budget and time. This paper deals with the progress of finding out best Software Cost Estimation method to give better accuracy on Software Cost Estimation

Key Words: Project Estimation, Effort Estimation, Cost Estimation, COCOMO & FPA – Function Point Analysis

Introduction:

Software cost estimation play an important role in software engineering, often determining the success or failure of contract negotiation and project execution. The main goal of software cost and effort estimation is to scientifically estimate the required workload and its corresponding costs in the life cycle of software system. Accurate cost estimates activity is critical to developers and customers. The overall process of developing a cost estimate for software is not different from the process for estimating any other element of cost. Some of the unique aspects of software estimating are driven by the nature of software as a product. Other problems are created by the nature of the estimating methodologies. Software cost estimation is a continuing activity which starts at the proposal stage and continues through the life time of a project. Continual cost estimation is to ensure that the spending is in line with the budget.

Accurate cost estimation is important for following reasons [1]:

- ✓ It can be used to classify and prioritize development projects with respect to complete business plan.
- ✓ It can help to find out what resources to commit to the project and how well these resources will be used
- ✓ It can help to assess the impact of changes and supporting for preplanning. Projects can be easier to manage and control when resources are matched to real needs.
- ✓ Customers expect accurate development costs to be in line with estimated costs

Related Work:

In current situation of software industries, successful project completion within time is most important task for any industry. The management point of view effort prediction is complicated task. Report says that around 65-80 percent of the project faces overrun of delivery date. Effort overrun directly proportional to cost overrun, so accurate effort prediction is also important. There are lot of many models available for prediction of software development effort and cost. COCOMO (Constructive Cost Model) [2] [3] is most commonly used model. But machine learning methods for software prediction are more appropriate, because they are more adaptable. When we are talking about specifically software development effort prediction problem, the output (effort value) of the system is very complexly dependent on input parameters, such as size of the problem, experience and many other. Now this complex relationship cannot be described or expressed using simple mathematical equations. In such situations neural network is more suitable to use, the reason can easily understand if we see at architecture of neural networks. Many researchers have applied the neural networks approach to estimate software development effort. Many different models of neural networks have been proposed. They may be grouped in two major categories. First one is feed forward networks where no loops in the network path occur. Another one is feedback networks that have recursive loops. The feed forward multilayer perceptron with back propagation learning algorithm are the most commonly used in the cost estimation field. Another study by Samson et al, uses an albus multilayer perceptron in order to predict software effort. They use Boehm's COCOMO dataset. Srinivasan and Fisher report the use of a neural network with a back propagation learning algorithm. They found that the neural network outperformed other techniques. Some primary work in the use of neural network in estimating software cost by Karunanithi et al. produced very

accurate results, but the major setback in their work was due to the fact the accuracy of the result relied heavily on the size of the training set.

COCOMO is arguably the most popular and widely used software estimation model, which integrates valuable expert knowledge. Understanding the adversity in applying neural networks, Nasser Tadayon has proposed a dynamic neural network that will initially use COCOMO II Model. COCOMO, however, has some limitations. It cannot effectively deal with imprecise and uncertain information, and calibration of COCOMO is one of the most important tasks that need to be done in order to get accurate estimations. So, there is always scope for developing effort estimation models with better predictive accuracy.

Estimation Techniques:

Generally, there are many methods for software cost estimation, which are divided into two groups: Algorithmic and Non-algorithmic [4] [5]. Using of the both groups is required 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 Methods (NA):

Contrary to the Algorithmic methods, methods of this group are based on analytical comparisons and inferences. For using the Non Algorithmic methods some information about the previous projects which are similar the under estimate project is required and usually estimation process in these methods is done according to the analysis of the previous datasets. Here, three methods have been selected for the assessing because these methods are more popular than the other None Algorithmic methods and many papers about their usage have been published in the recent years.

A. Expert Judgment Method: Expert judgment techniques involve consulting with cost estimation expert or a group of the estimation experts to use their experience and understanding of the proposed project to arrive at an estimate of its cost. It is the most usable methods for the cost estimation. Mostly companies used expert judgment method for generating the cost of the product. This method using following estimating steps: a. Project leader presents each expert with a specification and an estimation form. b. The experts fill out forms anonymously. c. Project leader calls a group meeting in which the experts discuss cost estimation issues with the project leader and each other. d. Project leader prepares and distributes a summary of the cost estimation on an iteration form. e. Again experts fill out forms, anonymously. f. Steps d and step e are iterated for as many rounds as appropriate.

B. Estimating by Analogy: Cost estimating by analogy means comparing the proposed project to previously completed similar project where the project development information is known. Actual data from the completed projects are extrapolated to cost estimate the proposed project. Analogy method can be used either at system level or at the component level. This method using following estimating steps: a. Find out the necessary characteristics of the proposed project. b. Choose the most similar completed projects whose characteristics have been stored in the historical data base. c. Find the estimate for the proposed project from the most similar completed project by analogy.

C. Top-Down Estimating Method: Top-down estimating method is known as Macro Model. Using this estimating method, an overall cost estimation for the project is derived from the global properties of the software project, and then the project is partitioned into various low-level mechanism or components. The method using this approach is Putnam model. Top-Down method is more applicable to early cost estimation when only global properties are known. In the early phase of the software cost estimation, top-down is very useful because there is no detailed information available.

D. Bottom-up Estimating Method: Using bottom-up cost estimating method, the cost of each software component is estimated and then combines the results to arrive at an estimated cost of overall project. Bottom-up method aims at constructing the estimate of a system from the knowledge accumulated about the small software components and their interactions. The method using this approach is COCOMO's detailed model.

Algorithmic Methods:

These methods are designed to provide some mathematical equations to perform software cost estimation. These mathematical equations are based on research and historical data and use some inputs for example Source Lines of Code, number of functions to perform, and some cost drivers like as language, design methodology, skill-levels, risk assessments, etc[6][7]. Algorithmic methods developed many models such as COCOMO models, Putnam model, and function points based models.

A. COCOMO Model: One very widely used algorithmic cost estimation model is the Constructive Cost Model (COCOMO) which was proposed by Boehm. The basic COCOMO model has a simple form:

MAN-MONTHS = K1* (KDLOC) ^{K}2 Where K1 and K2 are two parameters which are dependent on the application and development environment. Estimates from the basic COCOMO model can be made more accurate by taking into account other factors concerning the required characteristics of the software to be developed, the qualification and experience of the development team, and the software development environment. Complexity of the software has following factor:

- ✓ Reliability

- ✓ Data base size
- ✓ Required efficiency for memory and execution time
- ✓ Capability of analyst and programmer
- ✓ Team experience in the application area
- ✓ Experience of team in the programming language and computer
- ✓ Use of software engineering and tools

The first version of COCOMO model has been experiencing increasing difficulties in cost estimating of software developed to new life cycle processes and capabilities including rapid-development process model, reuse-driven approaches, object-oriented approaches and software process maturity initiative. COCOMO model is also serving as a framework for an extensive current data collection and analysis effort to further refine and calibrate the model's cost estimation capabilities.

B. Putnam Model: The Putnam model is an empirical effort estimation model. This model is very sensitive to the development time, decreasing the development time can greatly increase the person-months needed for development. One significant problem with this model is that it is based on knowing, or being able to estimate accurately, the size of the software to be developed. There is often great uncertainty in the software size. It may result in the inaccuracy of estimation.

C. Function Point Analysis: The Function Point Analysis is method of quantifying the size and complexity of a software system in terms of the functions that the systems deliver to the user. It was first introduced by Albrecht [8] [9]. The total number of FP depends on the counts of distinct in terms of format or processing logic types. Following two steps in counting function points:

- ✓ **Counting to the User Functions:** The raw function counts are arrived at by considering a linear combination of five basic software components. These components are external inputs, external outputs, external inquiries, logic internal files, and external interfaces, each at one of three complexity levels: simple, average or complex. The sum of these numbers, weighted according to the complexity level, is the number of FC.
- ✓ **Adjusting for Environmental Processing Complexity**: The final function points is arrived at by multiplying function count by an adjustment factor that is determined by considering 14 aspects of processing complexity. This adjustment factor allows the function count to be modified by at most 35% or -35%.

D. Linear Models: Commonly this models have the simple structure and trace a clear equation as below:

$$EFFORT = a0 + \sum a_i x_i$$

Where, a1, a2 ,an are selected according to the information of project, only allowed values for xi are -1, 0, +1.

Assessment of Cost Estimation Techniques:

At this section according to the previous presented subjects, it is possible to compare mentioned estimation methods based on advantages and disadvantages of them. This comparison could be useful for choosing an appropriate method in a particular project. On the other hand, selecting the estimation technique is done based on capabilities of methods and state of the project. Below table shows a comparison of mentioned methods for estimation. For doing comparison, the popular existing estimation methods have been selected [10] [11].

Table 1: Comparison among the Existing Methods

S.No	Method	Type	Merits	Demerits
1.	Expert judgment	Non-Algorithmic	Fast prediction, adapt for special projects	Success depend on expert, usually is done incomplete
2.	Estimation by analogy	Non-Algorithmic	Works based on actual experience and especial expert is not important	Much information about past projects is required in some situations there are no similar project
3.	Top-Down	Non-Algorithmic	Requires minimal project detail, usually faster and easier to implement and system level focus	Less detailed basis and less stable
4.	Bottom-up	Non-Algorithmic	More detailed basis, more stable and encourage individual commitment	May overlook system level costs , requires more effort, a lot of time consuming
5.	COCOMO	Algorithmic	Clear results, it"s very common	A lot of data is required, It is not suitable for any project
6.	Putnam model	Algorithmic	A Probabilistic model, it"s used in a very large project	For only use large projects
7.	Function Point	Algorithmic	Language free, its results are better than source line of	Mechanization is hard to do , it is not considered quality output

			code	
8.	Linear model	Algorithmic	It‟s a best method of prediction using linear regression technique	Little difference between actual and predicted results and error is also needed to calculate.

Conclusion:

In this paper, we provided a comprehensive overview of different types of software cost estimation methods and also describe 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. It has been seen that all estimation methods are specific for some specific type of projects. It is very difficult to decide which method is better than to all other methods because every method or model has an own significance or importance. The future work is to study new software cost estimation methods and models that can be help us to easily understand the software cost estimation process.

References:

1. Caper Jones” Estimating software cost “Tata Mc- Graw -Hill Edition 2007
2. Abedallah Zaid, Mohd Hasan Selamat, Abdual Azim Abd Ghani, Rodziah Atan and Tieng Wei Koh “Issues in Software Cost Estimation”, IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.11, November 2008
3. Pahariya, J. S., V. Ravi, et al."Software cost estimation using computational intelligence techniques". Nature & Biologically Inspired Computing, NaBIC 2009. World Congress on, 2009
4. Vahid Khatibi, Dayang N. A. Jawawi “Software Cost Estimation Methods: A Review” Journal of Emerging Trends in Computing and Information Sciences, Volume 2 No. 1, 2010-11
5. Venkatesh et al., International Journal of Advanced Research in Computer Science and Software Engineering 5(5), May- 2015, pp. 338-346
6. B. Boehm, and C. Abts, “Software Development Cost Estimation Approaches – A Survey1”, University of Southern California.
7. Shruti Jain, “Survey of Various Cost Estimation Techniques”, International Journal of Advanced Research in Computer Engineering and Technology (IJARCET), Volume-1, Issue-7, September, 2012.
8. Sweta Kumari and Shashank Pushkar,” Performance Analysis of the Software Cost Estimation Methods”, International Journal of Advanced Computer Science and Applications, Vol. 3, 2013.
9. Kusuma Kumari B.M, “Software Cost Estimation Techniques”, International Journal of Engineering Research in Management and Technology, Volume -3, Issue- 4,2014.
10. Liming Wu “The Comparison of the Software Cost Estimating Methods” University of Calgary
11. Shivangi Shekhar, Umesh Kumar “Review of Various Software Cost Estimation Techniques”, International Journal of computer Applications, Vol 141,2016

.