International Journal of Engineering and Technical Research (IJETR)
ISSN: 2321-0869, Volume-3, Issue-5, May 2015

# Software Cost Estimation: A Survey of Current Practices

**Ms. Aruna Shrivastava, Asst Prof. Leena Sahu**

***Abstract*— Today's software industry is all about efficiency. As technology trends are changing rapidly, implementation of more complex system at a cheaper cost & also maintenance of better quality at the same time are crucial challenges for software companies. Researchers have proposed numerous mechanisms for software cost estimation. This paper provides general overview of existing software cost estimation models and techniques. It also highlights strengths and weakness of various popular methods and covers the latest trends in this field..**

***Index Terms*—COCOMO , Estimation Process, Software Cost, SLOC, Software Metrics**

## I. INTRODUCTION

From the beginning of computer era, estimation of cost & effort involved in software development has been an important & challenging task. Software industry is getting more seasoned & complex these days because the size and importance of software applications have grown a lot. Without a doubt it is now the driving force of industry area, government & military operations, modern businesses, scientific, medical & technical fields. The vital link between the general concepts and techniques of economic analysis and the particular world of software engineering is provided by software engineering cost estimation. In order to make good management decisions and for accurately determining how much time, effort and resources are required, precise prediction of software development cost is must thus while software development one of the most crucial task is estimation. Many factors are responsible for accuracy of any project like size and level of complexities of project, business plans, resources required, resources used, impact of changes and re planning, customer expectations. According to studies most projects (60-80%) encounter effort and/or schedule overruns. Several models and techniques are available still accurate prediction is a challenge for analyst, software managers and researchers.

### *A. SOFTWARE COST & ESTIMATION PROCESS*

The process of predicting how many resources and how many hours are needed to develop a software project is called

SOFTWARE COST/ EFFORT ESTIMATION.
SOFTWARE COST comprises 3 major elements namely-

- Manpower
- Effort
- Duration

**Ms. Aruna Shrivastava**, Computer Science and Engineering, RCET, Bhilai (Chhattisgarh) India.
**Asst.Prof Ms. Leena Sahu**, Computer Science and Engineering, RCET, Bhilai (Chhattisgarh) India.

According to NASA'S guide on s/w cost estimation the basic equation for software cost is as follow (NASA, 2003):
Software development labor costs + other labor costs + Non-labor costs = Total Software costs
Software development labor costs – The labor, resources, time requires during different development phases like functional design, code development, requirement analysis, interface specification etc…

- Other labour costs – It covers costs involved during s/w quality assurance, test bed development, assembly, test & launch operations, verification & validation operations etc…
- Non-labor costs – Costs involved in training, travel & trips, software procurement, support & services etc…

### *B. Reasons for failure of IT Projects*

- Lack of knowledge & experience in estimation
- Lack of data, lack of time & resources to perform estimate accurately
- Poor user input and vague requirements
- Conflicts among stakeholders
- Rapid changes in IT and methodologies
- Poor architecture
- Improper planning, improper selection of SDLC model during development phases
  Poor risk management

### *C. Importance of Good Estimation*

Typically 4 major variables- time, requirements, resources (people, infrastructure/materials and money) and risks controls software projects. If any of these encounter unexpected changes, its impact will be on project. Both underestimation and overestimation of project needs can cause major problems.

- It can help to categorize and prioritize development projects according to an overall business plan.
- It is always helpful in assessing the impact of changes and helps in preplanning.
- When resources are better matched to real needs, projects can be easily managed and controlled.
- It helps in deciding what resources are needed to commit to the project and how well these resources will be used

## II. BACKGROUND

Before 1970, Thumb rules or some algorithms which were based on Trial and error were used for effort estimation [9].

1970 was a vital period to anticipate the expenses and schedules for software development. Computerized Software estimation tools were constructed. Some difficulties were experienced while building large software systems [16]. During mid 1970's the first automated s/w approximation tool had been flesh. The prototyping composite model is COCOMO (Constructive Cost Model) developed by Barry Boehm and is portrayed in book Software Engineering Economics [9]. In 1975, based on five different attributes namely-Inputs, Output, Inquires, Logical Files, Interfaces .A new Function Point Analysis approach was developed for estimation of size and development effort [2]. In 1977, Frank Freiman designed PRICE-S Software estimation model. In 1979, Lawrence H. Putnam introduced SLIM (Software Life Cycle Model) to US-Market [21].This model was based on Norden Rayleigh Curve. In 1983, Ada Programming language was introduced by DOD (U.S. Department of Defense). Ada-COCOMO model was build which reduced developing cost of large systems [22]. 1981, Dr. Barry Boehm highlighted the essential algorithms of Constructive Cost Model (COCOMO) through his book "Software Engineering Economics". During the same year Allan Albrecht published an article to the FPA method. This article sharpened the rules for rating the complexity of software [9]. In 1982, Tom de Marco introduced a functional metric that inherited some of the features of Albrecht's function point, but was developed independently. A book "controlling software projects" was released by him for introducing this metric. In 1983, Mark II function point metric was introduced by a British software estimating researcher Charles Symons [12]. In 1984 a major revision of function point metric was done by IBM which is basis of today's function points. 1985, In order to include the effect of computationally complex algorithms, concept of Function Point was extended by Caper Jones [3]. In 1986, IFPUG (International Function Point Users Group) was established in Toronto, Canada because of quickly developing utilization of Function Point Metrics. 1990, Barry Boehm, at college of Southern California started to revise and expand the idea of original COCOMO model. 1991 , Michel van Genuchten and Hans Koolen they added to various techniques and tools which were created over number of years to meet the expanding need to control programming advancement [10].1992, Betteridge, R. worked on software costing. A method called Mark II Function Point was used to predict cost of number of projects [8]. 1993, COCOMO 2.0 the new version of COCOMO was introduced which was emerged in 1994 [7]. 1994,Rajiv D Banker and Hsihui Chang and Chris F Kemerer, they thought that it was helpful for expense estimation and profit assessment purposes' to consider software development as a economic production process[4]. 1996, from the early system specifications Sophie Cockroft obtained accurate size estimations [14]. In 1997, techniques were more focused on accuracy and existing models were reviewed. In 1998, a new model called MARCS was constructed by Chatzoglou, to give predictions of the resources (time, effort, cost, and people) [13]. In 1999, J. J. Dolado, made a research using the technique of Genetic Programming (GP) for exploring possible cost functions [15]. In 2001, new approach was proposed which was based on reasoning by analogy and to estimate the effort linguistic quantifiers were used [1]. In 2002, M.Jorgensen, expert estimation was the most frequently applied estimation strategy for software projects [18]. In 2003, Yunsik Ahn, Jungseok Suh, Seungryeol Kim and Hyunsoo Kim, proposed SMPEEM (Software Maintenance Project Effort Estimation) [25]. In 2004,Barbara proposed the idea of EBSE (Evidence based Software Engineering) [6].In 2005, sequence was decided and needed to be carried out for software estimation Sizing Project deliverables, Estimating quality and defect Removal efficiency, Selecting Project activities, Estimating staffing levels, Estimating Effort, Estimating Costs, Estimating Schedules, Estimating requirements growth during development [11]. 2006, Stein Grimstad, effort estimate was frequently used without sufficient clarification of its meaning, and that estimation accuracy is often evaluated without ensuring that the estimated and actual effort were comparable[23].In 2007, for effort estimation different methods were introduced. The average accuracy of effort estimates based on expert judgment was higher than the average accuracy of models. In 2008, Parvinder S. Sandhu focused on predicting the accuracy of models. As a soft computing approach, neuro-fuzzy system was used to generate the model because Neuro-Fuzzy system was able to approximate the non-linear function with more precision [20]. 2010, In order to reduce the error and to minimize the changes of estimates from actual different estimation techniques were combined [19, 24]. 2011, numerous estimation techniques were proposed and used extensively by practitioners for use in Function Oriented Software development. 2012, A lot of commercial software costs estimating tools have been released till today..

## III. Review of Existing Methods

To improve the correctness of s/w development effort estimation several techniques and models have been proposed. These methods includes- Algorithmic estimation, analogy based estimation, data mining techniques, soft computing techniques, artificial neural network based techniques, expert judgment based techniques. This section enlists some of them along with their comparative advantages and disadvantages.

### *A. Algorithmic Estimation*

It uses mathematical equations to perform software estimations. These mathematical formulae relates independent variables( like cost drivers) to dependent variables (like effort, cost).Source lines of codes (SLOC),number of functions and other cost drivers such as languages, methodologies, risk assessment etc… are taken into account in this kind of estimation method. Model based on Algorithmic estimation are summarized in below table-

*TABLE I*

*Model based on Algorithmic estimation*

| Model | Effort Equation | Description |
|---|---|---|
| COCOMO Model | E= a ×(KLOC)*b × EAF | Developed by BOHEM, constant value a, b, depends on project type weather it is organic semi-detached or embedded. |

| | | |
|---|---|---|
| SLIM Model | Technical constant, **C = Size × B1/3 × T4/3** Total person months, **B=1/T4 × (Size/C)*3.** T=Development time in years. C= Parameter dependent on development environment. | It is empirical effort estimation model, developed by Lawrence H Putnam in 1978.It provides description of Time and effort needed to complete a software project of specified type. |
| Walston-Felix Model | E = 5.2 (KLOC) 0.91 D = 4.1 (KLOC) 0.36 | Developed by C.E.Walston and C.P. Felix in 1977. It is method of programming Measurement and estimation. |
| Albrecht-Gaffney MODEL | It uses Function point to estimate efforts. | Developed by IBM DP Services Organizations. |
| Kemerer MODEL | | It is cost estimation model uses Function points and Linear Regression. |

*B. Analogy based Estimation*

It compares new projects with similar projects from the past, make relationship and find similarity in order to find accurate result.

*C. Data mining Techniques*

These techniques transforms large and complex data into meaningful patterns and rules. Regression and Classification are some basic operation of data mining.

*D.Rule Induction*

It is particular aspect of inductive learning in which rules are produced by algorithms as a result of modelling. These rules are transparent and therefore can be read and understood easily, it is the advantage of inductive learning over neural network based learning.

*E. Artificial Neural Network based Estimation*

To find accurate estimate for software building efforts, Machine learning and Pattern Recognition methodology are used. ANN can learn from previous data and is able to find relationship between dependent and independent variables.

*F. Function Point analysis based Model*

At first in 1983, to measure the functionality of project Albrecht presented Function point analysis, this method measure the size of software, it considers internal logical files, external interface files, external input-output, external inquiries from functional viewpoint metric. ESTIMACS and SPQR/20 are the models which adopt FPA approach of estimation.

*G. Soft Computing Techniques*

Basically it is a consortium of methodologies cantering in artificial neural networks, fuzzy logic, and evolutionary computation. Particle Swarm Optimization, Ant Colony Optimization, Genetic Programming, fuzzy Logic etc… comes under this section, these methodologies provide in one form or another flexible information processing capability for managing real life ambiguous circumstances. These are complementary and synergistic, rather than competitive.

TABLE II
COMPARATIVE ADVANTAGES AND LIMITATIONS OF EXISTING METHODS

| METHODS | TYPE | KEY ADVANTAGES | DISADVANTAGES |
|---|---|---|---|
| COCOMO | Algorithmic | Very common approach provides clear results. | This model is not suitable for many projects as large amount of data is required. |
| Neural network based estimation methods | Non-algorithmic | These methods provide power of reasoning and are consistent with unlike databases. | Large training data is required. Lack of adequate amount of data set effects performance, no guidelines are available for designing. |
| Function point analysis | Algorithmic | Results are better than SLOC, Language independent, Since function points are based on system users external view of system, Non-tech users have a better Understanding of what FP are measuring. Resulting metrics are straightforward and logical. | Mechanization is hard to do as precise counting require In-depth knowledge of standards. |
| Analogy | Non-algorithmi | Having special experts is not | A lot of information about |

| | | | |
|---|---|---|---|
| based estimation | c | important, works based on actual experiences. | past similar projects is required. In some situations similar projects are not available. |
| Putnam model | Algorithmic | This model is basically based on 2 variables which are time and size. | This model does not consider all other aspects of software development life cycle. |
| Fuzzy logic based estimation | Non-algorithmic | This approach is capable to handle uncertainty and provides reliable estimates, no training is required, other advantage is its flexibility. | This method is hard to use, estimation of complex features cost is much tedious. |

## IV. Recently Developed Different Approaches and Findings

It has been observed that all estimation methods are specific for some specific type of projects. Every method or model has its own significance and importance therefore it is very hard to make a decision which method is better than to all other methods. This section summarizes the recent research, researchers worked with another field along with the software engineering like data mining and machine learning techniques for improving the accuracy of software cost estimation process.

*TABLE IIII*
*Different Cost Estimation Methods*

| **Researchers** | **Different Approaches For Software Cost Estimation And Findings** | |
|---|---|---|
| Witting and Finnie | In order to predict software development effort describe use of back propagation learning algorithms on a multilayer perception. | (26), (27) |
| Lefley and Shepperd | They applied the concept of genetic programming to improve software cost estimation on public datasets with great success | (28) |
| Prasad Reddy et al. | Used the concept of Multi Objective (MO) Particle Swarm Optimization and proposed a model for software cost estimation | (29) |
| Vinaykumar et al | For the prediction of software cost estimation used wavelet neural networks | (30) |
| Oliveira | This work is based on comparative study on support vector regression (SVR), radial basis functions neural networks (RBFNs) and linear regression for estimation of software project effort and result clears it that SVR significantly outperforms RBFNs and linear regression. | (31) |
| Pahariya et al | Described a new computational intelligence sequential hybrid architecture which includes programming and Group Method of Data Handling (GMDH). Data mining methods such as Radial Basis Function (RBF) Multi-Layer Regression (MLR), and so on are included in this work. | (32) |
| Reddy et al | By applying Gaussian membership function which provide better performance than the trapezoidal function to presenting cost drivers this work enhanced fuzzy approach for software effort of the COCOMO. | (33) |
| Andreou et al | This work considered Fuzzy Decision Trees (FDTs) for estimating required effort and software size in cost estimation as if strong evidence about those fuzzy transformations of cost drivers contributed to enhancing the prediction process | (34) |
| Sweta and Pushkar | This work provides a comparative study on Intermediate COCOMO, support vector regression (SVR and Multiple Objective Particle Swarm Optimization (MOPSO) model for prediction of project effort and it has been observed through simulation it has been observed that in comparison of other estimating techniques SVR provides better result in terms of accuracy and error rate. | (35) |

## V. Current Trends in Software Cost Estimation

### A. Use of SLOC/ SDI

The current trend is now trying to get away from SLOC/SDI and getting more focused on Function Points. The reason behind is that Function Points are more independent, less dependent on languages and programming environment as compared to SLOC/SDI.

### B. In House Metrics Development

Nowadays, the majority of systems developers and consultants have a methodology to find out the a priori cost of a software development project, such cost estimation methodology is allied to a specific systems analysis and design methodology. This estimation of cost is based on the use of the analysis methodology, knowledge and experience of the firm.

### C. Prototyping

Boehm and Papaccio's spiral development model is in essence a prototyping model in which a system is developed in phases, which includes requirements specifications, cost to completion, and the risk evaluated at each step. In recent years prototyping has become a major part of many systems developments efforts.

### D. Wide Commercial Industries of Estimation Tools

Estimation of cost/effort while software development is a complex activity, there is a commercial industry of companies which are marketing software estimation tools. As of 2013, most widely used tools for estimation purpose are-COCOMO II, SEER, SLIM, Software Risk Master (SRM), and TruePrice.

.

## VI. Conclusion and Future Work

A Process of estimation reflects the reality of project's progress. It manages cost/budget & controls overruns. No single method is necessarily better or worse than the other, in actual, strengths and weaknesses of each are often complimentary to one other .This paper provides a review of different types of methods in software cost estimation. To produce meaningful and reliable estimates, knowledge of each technique and understanding of software attributes and their casual relationship is must. More research is considered necessary to sizing the software functional requirement directly once it stores in CASE tool, which will result to quick estimation and reduction in cost. In object-oriented CASE environments, object points is one of the capable and promising technique but more research in this field is required.

.

## References

[1] Ali Idri, Alain Abran, Taghi M. Khosgoftaar. 2001. Fuzzy Analogy- A New Approach for Software Cost Estimation. *International Workshop on Software Measurement (IWSM'01)*.

[2] Allan J. Alberecht and John E. Gaffhey, November 1983, *Software Function, Source Lines of Code and Development Effort Prediction : A software Science Validation.* IEEE transactions on Software Engineering.

[3] Allan J. Alberecht, May 1984. *AD/M Productivity Measurement and Estimation Validation*, IBM Corporate Information Systems. IBM Corp.

[4] Banker, R. D., H. Chang, et al. (1994). *"Evidence on economies of scale in software development."* Information and Software Technology 36(5): 275-282.

[5] Barbara A. Kitchenham, Tore Dybå, Magne Jørgensen. 2004. IEEE Proceedings of the 26th International Conference on Software Engineering (ICSE'04).

[6] Barbara Kitchenham, Emilia Mendes. 2009. Why Comparative Effort Prediction Studies may be Invalid © ACM 2009 ISBN: 978-1-60558-634-2.

[7] Barry W. Boehm, Bradford dark, Ellis Horowitz, Chris Westland, Ray Madachy and Richard Selby. Cost Models for Future Software Lifecycle Processes: COCOMO 2.0 Annals of Software Engineering. Volume 1, pp 57-94, 1995. An earlier description was presented in the tutorial *"COCOMO, Ada COCOMO and COCOMO 2.0"* by Barry Boehm in the Proceedings of Ninth International COCOMO Estimation Meeting. Los Angeles, CA, 6-7 October 1994.

[8] Bergeron, F. and J. Y. St-Arnaud (1992). *"Estimation of information systems development efforts: a pilot study."* Information and Management 22(4): 239-254.

[9] Boehm, 1981 *"Software Engineering Economics",* Prentice Hall.

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

[11] Capers Jones, Chief Scientist Emeritus Software Productivity Research LLC. *How Software Estimation Tools Work.* Version 5 – February 27, 2005.

[12] Charles Symons 1991. *Software Sizing and Estimation Mark II function Points (Function Point Analysis),* Wiley 1991.

[13] Chatzoglou, P. D. and L. A. Macaulay (1998). *"A rule-based approach to developing software development AA prediction models."* Automated Software Engineering 5(2): 211-243.

[14] Cockcroft, S. (1996). *"Estimating CASE development size from outline specifications."* Information and Software Technology 38(6): 391-399.

[15] Dolado, J. J. (2000). *"A validation of the component-based method for software size estimation."* IEEE Transactions on Software Engineering 26(10): 1006-1021.

[16] F.Brooks, The Mythical Man-Month; Essays on Software Engineering, 1975. Addison-Wesley, Reading, Massachusetts.

[17] Lawrence H. Putnam 1978. *A General Empirical Solution to the Macro Software Sizing and Estimation problem.* IEEE transactions on Software Engineering.

[18] Magne Jørgensen, *A Review of Studies on Expert Estimation of Software Development Effort*, March 2002.

[19] M. V. Deshpande, S. G. Bhirud. August 2010. *Analysis of Combining Software Estimation Techniques.* International Journal of Computer Applications (0975 – 8887).

[20] Parvinder S. Sandhu, Porush Bassi, and Amanpreet Singh Brar. 2008. *Software Effort Estimation Using Soft Computing Techniques.* World Academy of Science, Engineering and Technology 46 2008.

[21] 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.

[22] Robert C. Tausworthe, 1981. Deep Space Network Estimation Model, Jet Propulsion Report.

[23] Stein Grimstad, Magne Jørgensen, Kjetil Moløkken-Østvold. 13 June 2005. *Software effort estimation terminology: The tower of Babel.* Information and Software Technology 48 (2006) 302–310.

[24] Vahid Khatibi, Dayang N. A. Jawawi. 2010. *Software Cost Estimation Methods: A Review.* Journal of Emerging Trends in Computing and Information Science.

[25] Yunsik Ahn, Jungseok Suh, Seungryeol Kim and Hyunsoo Kim. July 2002. Journal of Software Maintenance and Evolution: Research and Practice.

[26] H. Agahi, S. Malhotra and J. Quirk, Estimating Software Productivity and Cost for NASA Projects, Journal of Parametrics, pp. 59-71, 1998.

[27] S. Chulani, B. Boehm and B. Steece, From Multiple Regression to Bayesian Analysis for Calibratuing COCOMO, Journal of Parametrics, vol. 15(2), pp. 175-188, 1999.

[28] M. Lefley and M. J. Shepperd, *Using Genetic Programming to Improve Software Effort Estimation Based on General Data Sets*, LNCS, Genetic and Evolutionary Computation — ISBN: 978-3-540-40603-7, page-208, GECCO 200.

[29] Prasad Reddy P.V.G.D, Hari CH.V.M.K and Srinivasa Rao, *Multi Objective Particle Swarm Optimization for Software Cost Estimation*, International Journal of Computer Applications, Vol.-32, 2011.
[30] K. Vinaykumar, V. Ravi, M. Carr and N. Rajkiran, *Software cost estimation using wavelet neural networks*, Journal of Systems and Software, pp. 1853-1867, 2008.
[31] Andriano L.I. Oliveira, *Estimation of Software Project Effort with Support Vector Regression*, www.journals.elsevier.com/neurocomputing, Vol.- 69, Issues 13–15, pp. 1749–1753, August 2006.
[32] J. S. Pahariya, V. Ravi, M. Carr, M. Vasu, *Computational Intelligence Hybrids Applied to Software Cost Estimation*, International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM),Vol. 2, pp. 104-112, 2010.
[33] C. S. Reddy, K. Raju, *An Improved Fuzzy Approach for COCOMO's Effort Estimation using Gaussian Membership Function*, Journal of Software, VOL. 4, NO. 5, pp. 452-459, 2009.
[34] A. S. Andreou, E. Papatheocharous, *Software Cost Estimation using Fuzzy Decision Trees*, 23rd IEEE/ACM International Conference on Automated Software Engineering, pp. 371 - 374, 2008.
[35] Sweta Kumari and Shashank Pushkar, *Comparison and Analysis of Different Software Cost Estimation Methods*, International Journal of Advanced Computer Science and Applications, Vol. 4, No.1, 2013

**Ms. Aruna Shrivastava** has received the B.E. degree from Chhattisgarh Swami Vivekanand Technical University, Bhilai (C.G.) India in Computer Science & Engineering in the year 2012. She is currently pursuing M.Tech. Degree in Computer Science Engineering with specialization in Software Engineering from CSVTU Bhilai (C.G.), India. Her research area includes Data Mining

**Ms. Leena Sahu** is currently Assistant professor in Department of Computer science & Engineering RCET, Bhilai (C.G.) India. She completed her B.E and M.Tech.Computer Science and Engineering Branch. Her research area includes Data Mining, Computer Network etc. She has published many Research Papers in various reputed National & International Journals, Conferences and Seminars.