

Systematic Literature Review of Identifying Issues in Software Cost Estimation Techniques

Muhammad Asif Saleem¹, Tahir Alyas³, Asfandayar⁵

Department of Computer Sciences
Lahore Garrison University
Lahore, Pakistan

Muhammad Idrees⁴

Department of Computer Sciences & Engineering
The University of Engineering and Technology
Narowal Campus, Pakistan

Rehan Ahmad², Asif Farooq⁶, Kahawaja Ali⁸

Department of Computer Science
The University of Lahore, Lahore, Pakistan

Adnan Shahid Khan⁷

Faculty of Computer Sciences and Information Technology
Universiti Malaysia Sarawak, Sarawak, Malaysia

Abstract—Software cost estimation plays a vital role in software project management. It is a process of predicting the effort and cost in terms of money and staff required for developing the software system. It is very much clear that software project will be successful if its estimated cost will be near to the real cost. When the project is at the acquisition stage the least details are available about a software project to be developed, due to which Problems arises in cost estimation. As the stages move on details increases for software development of software which is quite fruitful in cost estimating. However, it can be considered that estimating the software cost in the first phases will produce better results. In this research cost estimation techniques are discussed along with the issues in that particular technique and focus will be on understating the points or issues which cause hurdles or issues in estimating the cost of the software project.

Keywords—Cost estimation; COCOMO; qualitative; use case point, PCA

I. INTRODUCTION

Estimating the software cost is very important but it is difficult to do so as well. In the early era of software development, there were very fewer instructions in software, as the software size is increased the accuracy in the cost of estimation also increased. Nowadays large software projects increased up to 25 million lines of source code [1]. Which may need 1000 software developers to code which may take 5 years to complete. The cost of that project may increase to 400 million dollars. So mistakes in cost estimation of these types of project will be very ample serious indeed. A huge amount of large software systems could not meet the deadlines, runs over budget or face cancellation of the project due to underestimation or overestimation of software cost at the phase of requirement engineering. In fact, overconfidence in measuring the software cost is a vital cause of expansion in software budget, failures, and litigation. Now a day's software industry is imparting the main role in running every type of business in the public and private sector. It can be considered that the government of a state or a private corporation builds thousands of software applications and may change them according to their needs every year. So being a welcome agent of a software project in this world, every software house who

dealing in large or complex software system development pays huge focus on estimating the cost of the project which is going to be developed. In this study different software cost estimation techniques proposed and finding their issues with the help of systematically selecting the researching databases, selecting article and deciding their selection or rejection criteria. This study will help the researchers as well as practitioners for selecting suitable software cost estimation model. For this systematic literature study article is divided into multiple sections which are as:

- Methodology for literature study
- Literature Study
- Software cost estimation techniques and Disadvantage
- Issues and problems in software cost estimation models
- Final Conclusion and future work

II. METHODOLOGY FOR LITERATURE STUDY

The systematic literature review has been utilized to do this research. It is an appropriate and repeat procedure to record relevant points of interest in the exact research range for inspecting and examining all current research identified with research questions. Thus, this exploration consolidates following phases like Categories definition, Review protocol development, and selection and rejection criterion and Search process.

A. Category Definition

We have characterized six categories with a specific end goal to sort out the search result. This order will fundamentally manage the correctness of the exact responses. The brief description is given below.

B. General Category

There may be various studies in software cost estimation along with different modular approaches. (Expenses on modeling, transformation, verification, and simulation). As per our survey study, we have categorized the software cost estimation in the categories like Algorithmic approach, Non Algorithm approach, and Expert opinions.

TABLE. I. SEARCH PROCESS

	No. of Articles							
	Library	Operator	Library	Operator	Library	Operator	Library	Operator
Keywords	IEEE	AND	ACM	AND	SPRINGER	AND	ELSEVIER	AND
Software cost estimation	955	32	3154	29	327	3	46,358	2
Software Cost determination	97	2	3055	1	102	0	4,384	0
Software effort estimation	454	55	2774	37	432	10	5814	13
Software cost calculation	246	8	2764	0	330	0	1169	0
Software cost detection	1037	1	3,205	0	469	0	9140	0
Tools software cost	1586	1	3,246	1	868	0	2678	1
Framework software cost	1314	1	3,698	0	855	0	16206	0
Models Software Cost estimation	4670	12	5,051	0	1164	0	8989	3
Methods software cost	3169	4	4,577	1	1111	0	25534	0
Techniques software cost	2332	3	4,651	0	1086	0	21216	0
Software cost estimation steps	525	4	2,989	0	1009	0	7795	0
Software cost estimation phases	740	1	2,815	0	687	0	4,745	0
Total	17125	123	41979	68	8008	13	148213	19

C. Review Protocol Development

As the categories for software costing is defined our next phase is to develop a protocol for searching the relevant articles. For review protocols, we have considered search process, selection and rejection criteria, quality assessment and data extraction.

D. Search Process

As the table shows above, it shows that we have chosen four logical databases (i.e. IEEE, ELSEVIER, SPRINGER, and ACM) with a specific end goal to complete our research. These logical databases maximum result regarding our topic and we have selected the required article via the self-defined process described below.

We have selected year-wise range from "2013–2017" for selection of research articles from our selected logical databases. We use different keywords for selecting articles (e.g. SCE, SCDE), different operators like AND OR are also used for making our search accurate. We have applied AND operator in abstract, keywords and title of the article. In this way the result from logical databases are refined i.e. non-relevant articles are removed in this way. The complete search process is defined and practices given in Table I.

III. SELECTION CRITERIA

We characterize the solid paradigm for the choosing and rejecting of research works. Six parameters are characterized to guarantee the correctness of the appropriate responses of our research questions. The research work will be chosen on the premise of these parameters as given underneath.

A. Subject-Relevant

Select the research work just if that it comes fits our research settings made by our category design. It must encourage the appropriate responses of our research addresses and should be applicable to one of the predefined

categorizations. Dismiss insignificantly explores those don't have a place with any of the predefined categories.

B. Year Wise Range 2013-2017

Chosen research work must be distributed from 2013 to 2017 by inserting filter an off year-wise selection in every database which will result reduce overall results and helps in finalizing the research article. Reject all the article less our selection criteria set in year wise selection mechanism.

C. Publisher

Selected research work must be published in one of the four renowned scientific databases i.e. IEEE (IEEE Scientific Database, 2014), SPRINGER (Springer, 2014), ELSEVIER (Elsevier, 2014) and ACM (ACM, 2014).

D. Crucial-Effects

Chosen research work must have vital beneficial results with respect to software cost estimation under software project management approach. Dismiss the research work if its proposal does not have a large contribution estimation of software cost.

E. Results-Oriented

Selected research work must be results-oriented means producing a result, not a simple survey. Have a brief look on the result section of the selected article and verify that that contribution of this article has major worth in the field of software cost estimation. Result verification must be carried out by a powerful survey if it does not so dismiss the work.

IV. REJECTION CRITERIA

A. Repetition

All the research in a specific research setting can't be incorporated. Thus, dismiss the research if these are indistinguishable in the given research setting and just a single of them is chosen.

B. Rejection on the Title Bases

Selected research work can be justified by having a brief look at the title of the research articles. It may need some expertise while judging the article but it will provide a fruitful result. The proposition and extreme results of the research must be upheld by strong certainties and experimentation. Dismiss the work if its title is not matching you're related to the research topics.

C. Rejection on Basis of Abstract

Sometimes it is very much hard to take decision while selecting the research article by checking with the title of the article so in this regard you should read the abstract of the article from which you can get proper information regarding the article.

V. LITERATURE STUDY

Software costing is important to phase in software project management [1]. It is also crucial and difficult for Development Company and client as well. Software cost estimation is essential to the contract, negotiating with the client and writing for the request of the proposal, Monitoring and controlling. Overestimation in cost may result in maximum resources promised to the project or bidding in contract, due to all these issues project may not win for the company. Not winning the project may lead to termination to the job. Accurate cost estimation is important because:

- as per the overall plan of business, it may be helpful in the classification and prioritization of development project [2];
- accurate cost estimation may be helpful in calculating the resources for committing to the project;
- for the impact of change and support preplanning;
- it is better to manage software project when resources areas the needs; and
- customers are always in view that the real cost of software should be equal to estimated costs.

Measuring software cost depends upon on from these factors:

- Software Effort (Time, resources)
- Duration of project (as per calendar time)
- Software Cost (Amount)

Experts have studied these three problems given below:

- Which model of software cost estimation is to use?
- Which software size measurement to use lines of code (LOC), function points (FP), or feature point?
- What is a good estimate?

VI. EXPERT JUDGMENT

In this method, we consult with an expert in software cost estimations and a group of experts who have experience in this field [3, 4]. By using their expertise and by understanding the

nature of the project we came at the point to estimate the cost of the project.

This method has several weak points which are listed below:

- Non-quantifiable.
- Documenting the factor taken from an expert is difficult.
- Maybe the opinion of experts be biased [5].
- As the total information is not available at the initial phase so the proposed estimation may lead to the wrong [6].
- This process may be time-consuming because of lots of discussion among experts [7].
- Due to personal limitation or time may this method not be feasible.

VII. ALGORITHMIC

In the algorithmic cost estimation methods, mathematical approaches are used for the purpose of estimation [8]. These mathematical approaches are based on historical data and research, which are used as inputs in function points, source lines of codes and other cost drivers. The algorithmic models are studies on large scale like the Putnam Model, COCOMO model and function point models.

- Issues in the algorithmic model are as follows:
- In this method dealing with exceptional conditions [5].
- False cost driver ratings and poor sizing inputs will produce poor costing results [9].
- Some factors may not be quantified resulting in bad costing results.

VIII. COCOMO

COCOMO Model is known as cost constructive model [10]. This model is widely used model in branch of cost estimation of software. Basic COCOMO Model has very simple form which can be described as:

$$\text{MAN+MONTH} = \text{KI} * (\text{delivered source instructions})^{\text{K2}}.$$

Here k1 and K2 are the parameters which are completely dependent on the environment of development and application under process. Estimation in the COCOMO model can be more precise if the following factors are kept in mind while following the steps of COCOMO models [2]. Characteristics of software under development, Qualification, and experiences of team members, Environment to be used for development purpose. Some of these factors are:

- Complexity of Software
- Reliability of software
- Size of database
- Efficiency required

- Teams experiences involved in the development
- The capability of developers involved in the analysis process
- Experience of software team with respect to computer and programming languages which are going to use in development purpose.

Many of the above issues may affect the MAN and MONTH required in order. COCOMO model can perform well if the requirement of software to be developed might be clear and stable. Which is mostly not clear or stable due to which its efficiency in these types of projects is not up to marks. This model is the regression model as it is based on 63 selected papers [9]. Primary inputs for this model is KDSI. Problems and issues in these models are described as:

- At the initial phase, there are lots of uncertainty in cost estimation so it is not possible to get near to the exact value of cost.
- COCOMO model is resulted by having the analysis of 63 selected papers. It usually has some problems at environment due to which recalibration is mandatory.

As per Kemmerer's research, the average bug rate in maximum versions of the model is recorded 601% [11]. The first version of the COCOMO model was developed in 1981. Nowadays it is facing problems in estimating the cost 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. Due to all these issues faced in this model new approach is designed which is named as COCOMO II.

Disadvantages in this models are described as follows:

- This model completely forgets requirement and all documentation [12]
- It also forgets customer's skills like knowledge, cooperation, and other parameters
- It didn't focus on the impact of security and safety
- It ignores personnel turnover level
- It doesn't consider hardware issues
- It is dependent on time spent in every phase.

IX. PUTNAM MODEL

This model is also algorithmic model developed after COCOMO II [13]. It is very much popular model with respect to cost estimation.

Disadvantages of this model are given as under.

This model is dependent on the size of the software to develop by knowing it or being able to estimate it correct. As we know there is a factor to great uncertainty in knowing the size of the software. It is understood that if the size of the software is not accurate than the cost estimated by the Putnam

model will not be correct. As per Kemmerer research Putnam model is based on SLIM which is 773.00% [1].

X. FUNCTION POINT METHODS

Function point methods method was created by Allan Albrecht at IBM and this model published in 1979 [10, 14]. This model has many advantages over source line of code method count of sizing. The function point data collection has two major motives firstly as per the desire of the manager to check out the level of productivity. Secondly, it is used in the estimation of cost.

Problems or issues in this model are briefly described as [3]:

- This model works on the subjective approach with having maximum judgments considered.
- Lots of models are like effort or cost estimation is based on LOC so this should also be converted.
- Amount of data research is very much less in this model.

XI. ANALOGY BASED

Estimation in this analogy based model is carried by comparing the proposed project with already developed projects with same requirement [15]. For this purpose, data is extracted from the already extracted project and compared with the proposed project. This methodology can be used on the project level as well as a component level [16].

Estimating by analogy is straight forward [2]. it is a very efficient way of expert judgment when experts were often in search for analogous situations so as to inform their opinion [17].

Disadvantages in these methods are as follows:

- Selection of projects for getting data from already developed is quite a difficult task [18].
- We are deriving an estimate for the new project with the help of already developed values from the randomly selected projects. Possible situations may involve means and weighted means which may provide more impact on the near to analogies.

XII. TOP-DOWN METHODS

The top-down strategy is also known as a macro method. In this method, the project cost is estimated from the universal properties of the project [19]. Then the project is divided into sub-projects or software component. This method can be used at the early phase of development because at that stage minimum details are available which can be tackled in this model easily [20]. This method is useful when no detailed information is there.

Disadvantages of these methods are as follows:

- This model mostly ignores the low-level problems in software cost estimation that may affect the estimation of software cost [21].

- This model is unable to provide a detailed basis of decisions or estimations.

XIII. BOTTOM-UP

In the bottom up [22] approach every single component's cost is measured and then at the end total cost is measured by calculating every component's cost. The most leading model which is using this approach is COCOMO [19, 23].

The problems or issues in these methods are described as follows:

- Cost estimated in this method may not be correct because most of the details are not available at early phases [24].
- This method is time-consuming.
- This method may not be able to use when the resources are limited.

XIV. DISCUSSION

Costing for software projects have distinctive elements that make the cost estimation so hard [25]. The complex ranges that impact the cost estimation are:

- Lacking in abilities of Software and maximum utilization of already available skills.
- Optimum utilization of COTS parts.
- Estimation of software.
- System of frameworks, specifically security issues.
- Interoperability.
- Catching of Requirements.
- Obsolescence.
- Designing software.

Estimating Software cost is so difficult in light of the fact that:

- The software truly is substantially difficult. Consider the number of requests for greatness included. Time scale 24x7 down to 100 nanosecond clock cycles is a LOT of requests of extent: beyond any reasonable amount to completely fathom. Henceforth the innate trouble. Consequently the vulnerability.
- Because you have a point where the project should leave meanwhile client is unable to understand how the clients fulfill its need [26].
- The major fault at client side that the client does not focus on "what he needs" More awful, they can't know until the point when they begin to work with the arrangement innovation which at that point changes the idea of the issue [26].
- We focus on the upcoming by taking a gander at the past. So I think there are two principle issues with estimation [27].

- If we were made a request to finish the something we have never done, we can't know to what degree, it will take since we have never performed it sometime in the near future.
- If we do not manage a record of what extent matter may take place, even when we're made a request to accomplish something comparable once more, we won't have the information whereupon to gauge.

XV. CONCLUSION

From this system literature study following conclusion are drawn:

- The cost estimation is a complex task and all the methods proposed have multiple advantages and disadvantages. Like some models performs better in large projects, some performs better on small projects and some performs better on global software development.
- None of the factors, which may affects cost estimation be ignored while estimation cost of software.
- For best and consistent cost estimation, the predictive attributes from the data set may considered. Backward input approach maybe use for selection these attributes.
- In this study, multiple methods are considered, the best efficiency is provided by collaborative methods.
- For better results, Experts opinion must be kept in mind. Biasness of experts may judge and removed.

The future work in cost estimation is proposing a novel model using hybrid approach. Which will be further enhanced by expert's opinion.

REFERENCES

- [1] P. Agrawal and S. Kumar, "Early phase software effort estimation model," in Colossal Data Analysis and Networking (CDAN), Symposium on, 2016, pp. 1-8.
- [2] S. Bilgaiyan, S. Mishra, and M. Das, "A review of software cost estimation in agile software development using soft computing techniques," in Computational Intelligence and Networks (CINE), 2016 2nd International Conference on, 2016, pp. 112-117.
- [3] W. Han, T.-B. Lu, H. Jiang, and W. Li, "A Study on the Improvement of Software Effort Estimation," in Control and Automation (CA), 2014 7th Conference on, 2014, pp. 24-27.
- [4] M. El Bajta, A. Idri, J. L. Fernández-Alemán, J. N. Ros, and A. Toval, "Software cost estimation for global software development a systematic map and review study," in Evaluation of Novel Approaches to Software Engineering (ENASE), 2015 International Conference on, 2015, pp. 197-206.
- [5] K. Usharani, V. V. Ananth, and D. Velmurugan, "A survey on software effort estimation," in Electrical, Electronics, and Optimization Techniques (ICEEOT), International Conference on, 2016, pp. 505-509.
- [6] M. Saroha and S. Sahu, "Software effort estimation using enhanced use case point model," in Computing, Communication & Automation (ICCCA), 2015 International Conference on, 2015, pp. 779-784.
- [7] P. Pandey, "Analysis of the techniques for software cost estimation," in Advanced Computing and Communication Technologies (ACCT), 2013 Third International Conference on, 2013, pp. 16-19.
- [8] L. V. Patil, R. M. Waghmode, S. Joshi, and V. Khanna, "Generic model of software cost estimation: A hybrid approach," in Advance Computing Conference (IACC), 2014 IEEE International, 2014, pp. 1379-1384.

- [9] M. Phongpaibul and P. Aroonvatanaporn, "Standardized cost estimation in Thai government's software development projects," in *Computer Science and Engineering Conference (ICSEC), 2015 International*, 2015, pp. 1-6.
- [10] S. Islam, B. B. Pathik, M. H. Khan, and M. M. Habib, "A novel tool for reducing time and cost at software test estimation: An use cases and functions based approach," in *Industrial Engineering and Engineering Management (IEEM), 2014 IEEE International Conference on*, 2014, pp. 312-316.
- [11] N. Garcia-Diaz, C. Lopez-Martin, and A. Chavoya, "A Comparative Study of Two Fuzzy Logic Models for Software Development Effort Estimation," *Procedia Technology*, vol. 7, pp. 305-314, 2013.
- [12] J. Huang, Y.-F. Li, and M. Xie, "An empirical analysis of data preprocessing for machine learning-based software cost estimation," *Information and Software Technology*, vol. 67, pp. 108-127, 2015.
- [13] L. L. Minku and X. Yao, "Ensembles and locality: Insight on improving software effort estimation," *Information and Software Technology*, vol. 55, pp. 1512-1528, 2013.
- [14] H. Rastogi, S. Dhankhar, and M. Kakkar, "A survey on software effort estimation techniques," in *Confluence The Next Generation Information Technology Summit (Confluence), 2014 5th International Conference-*, 2014, pp. 826-830.
- [15] S. Garg and D. Gupta, "PCA based cost estimation model for agile software development projects," in *Industrial Engineering and Operations Management (IEOM), 2015 International Conference on*, 2015, pp. 1-7.
- [16] D. Toka and O. Turetken, "Accuracy of Contemporary Parametric Software Estimation Models: A Comparative Analysis," pp. 313-316, 2013.
- [17] N. Mittas, E. Papatheocharous, L. Angelis, and A. S. Andreou, "Integrating non-parametric models with linear components for producing software cost estimations," *Journal of Systems and Software*, vol. 99, pp. 120-134, 2015.
- [18] Y.-S. Seo, D.-H. Bae, and R. Jeffery, "AREION: Software effort estimation based on multiple regressions with adaptive recursive data partitioning," *Information and Software Technology*, vol. 55, pp. 1710-1725, 2013.
- [19] S. Al-Qudah, K. Meridji, and K. T. Al-Sarayreh, "A Comprehensive Survey of Software Development Cost Estimation Studies," pp. 1-5, 2015.
- [20] A. Ren and C. Yun, "Research of Software Size Estimation Method," pp. 154-155, 2013.
- [21] F. Sarro, A. Petrozziello, and M. Harman, "Multi-objective software effort estimation," pp. 619-630, 2016.
- [22] D. Kashyap and A. K. Misra, "Software development cost estimation using similarity difference between software attributes," presented at the *Proceedings of the 2013 International Conference on Information Systems and Design of Communication*, Lisboa, Portugal, 2013.
- [23] C. S. Stokes, A. R. Simpson, and H. R. Maier, "A computational software tool for the minimization of costs and greenhouse gas emissions associated with water distribution systems," *Environmental Modelling & Software*, vol. 69, pp. 452-467, 2015.
- [24] A. Girasella and F. Pagnin, "An UML-based approach to software development cost estimation," presented at the *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, Torino, Italy, 2014.
- [25] M. Jørgensen, "Communication of software cost estimates," pp. 1-5, 2014.
- [26] T. Urbanek, Z. Prokopova, R. Silhavy, and V. Vesela, "Prediction accuracy measurements as a fitness function for software effort estimation," *Springerplus*, vol. 4, p. 778, 2015.
- [27] L. Kompella, "Advancement of Decision-Making in Agile Projects by Applying Logistic Regression on Estimates," pp. 11-17, 2013.