Software Effort Estimation: A Survey of Well-known Approaches

Amid Khatibi Bardsiri
Computer Engineering Department, Bardsir Branch, Islamic Azad University, Kerman, Iran
E-mail: a.khatibi@srbiau.ac.ir

Seyyed Mohsen Hashemi
Assistant Professor, Science and Research Branch, Islamic Azad University, Tehran, Iran

Abstract

Project Failure is the glaring issue considering today while observed by software experts. The imprecision of the estimation is the reason for this challenge. Software effort estimation is the major fundamentals of software development. It truly is the liability of software project management ; that he basically manages the financial plan challenges and also ought to handle the entire task within the assigned hour slot. Large amount of estimation techniques , models and also approaches are utilized; but yet not any of those can easily supply 100% precision in cost , in time or else in any further estimation aspects. Precise estimation is a sophisticated procedure since it might be visualized like software effort forecast, since the phrase shows foresight not ever will become a reality. Effort estimation usually needs generalizing from a few old projects. Generalization from these kinds of restricted knowledge is an naturally under light situation. Variety of participants make their activities to generate different methods in last three decades. This article is related to the extensive descriptive discovery of the models which are introduced in the beginning of the software estimation area in addition to includes many of the well-known accessible and utilized parametric models or number of non-parametric methods. Furthermore evaluating the software estimation tactics detailed, and produces the choice of suitable estimation model simpler. The major summary is this not any unique approach is best for most circumstances, which an attentive assessment of the information on a number of methods might be to generate practical estimates.

Keywords: Software Effort Estimation, Survey, Metrics, Models, Approaches

1. Introduction

Software cost and effort estimation is truly a very important and elaborate, however a necessary task in the software development procedures. During the last 3 decades, an increasing tendency has been seen in a utilizing number of software estimation techniques in numerous computer companies. In addition to this enormous progress, it is usually made real the essentiality of most such techniques of estimation the software efforts and also making plans quicker or simply in the estimated situations. However lots of investigation hours, or funds has been spent in enhancing precision of the different estimation products, because of the natural risk in software development efforts similar to elaborate and changing communication aspects, inherent software sophistication, stress on standardization and also insufficient software information, it will be not realistic to calculate quite a precise effort estimation of software [1]. The precision of the specific models chooses their usability in the planned areas, while the accurateness may be described driven by knowing the calibration of the software information. Because the accuracy and so credibility of the software effort estimation is critical to the competitiveness of computer organizations, the analysts have put their fullest energy to create the correct models for estimate cost close to true wrongs. There are a lot of techniques have already been suggested that will be classified based upon their core composition strategies; analogy based estimation [2], estimation by expert [3], rule induction techniques [4], algorithmic techniques with empirical strategies [5], artificial neural network designed methods [6, 7, 8], decision tree based strategies [9], Bayesian network techniques [10] and fuzzy logic based estimation structures [11, 12]. Cost and Effort are tightly relevant, these are not always connected by an easy conversion operation. The effort is usually calculated in person-months of the managers, analysts and also programmers so on. Choosing a model while the ideal appears to be not possible due to the fact the efficiency of every model relies on numerous aspects like accessible data, development methods, project characteristics etc. Thus, estimate of software metrics might be more complex versus various other things. Therefore an intensive recognize along with an attentive evaluation is needed for the assessment of every technique to be able to produce ideal process which can result in the most precise and reasonable estimates. Minor Projects are really simple to estimate and so precision may not be crucial. However since the scale of the project raises, needed precision is essential that is difficult to estimate. A great calculation must have a level of granularity then it may be described. As the effort acquired a project is among the primary and so many assessed factors. Estimating the effort with a significant value of credibility is an issue not already having been overcome still. Since may be seen from Figure 1, the natural trouble with estimating is usually that small projects are often very simple

to estimate, however the needed precision is not crucial . Alternatively, big projects have become not easy to estimate, nevertheless the needed precision is essential.

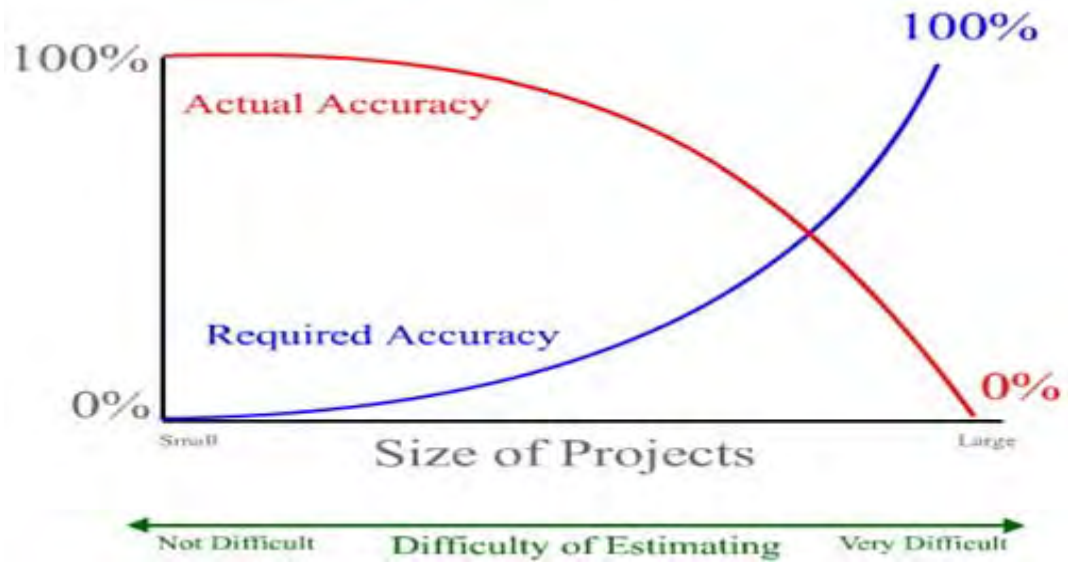

Figure 1: Accuracy of estimating

While the project continues on the project ought to be reversed. I.e., as increasingly more details are recognized relating a project the estimate needs to be modified. It is essential to establish upper and lower bounds for a project estimations. Figure 2 shows the way estimates have to merge with the real while the project continues. This article is the short presentation of software estimation metrics, an extensive descriptive research on the estimation models which introduced at the beginning of the software estimation area and then addresses many of the well-known accessible or utilized.

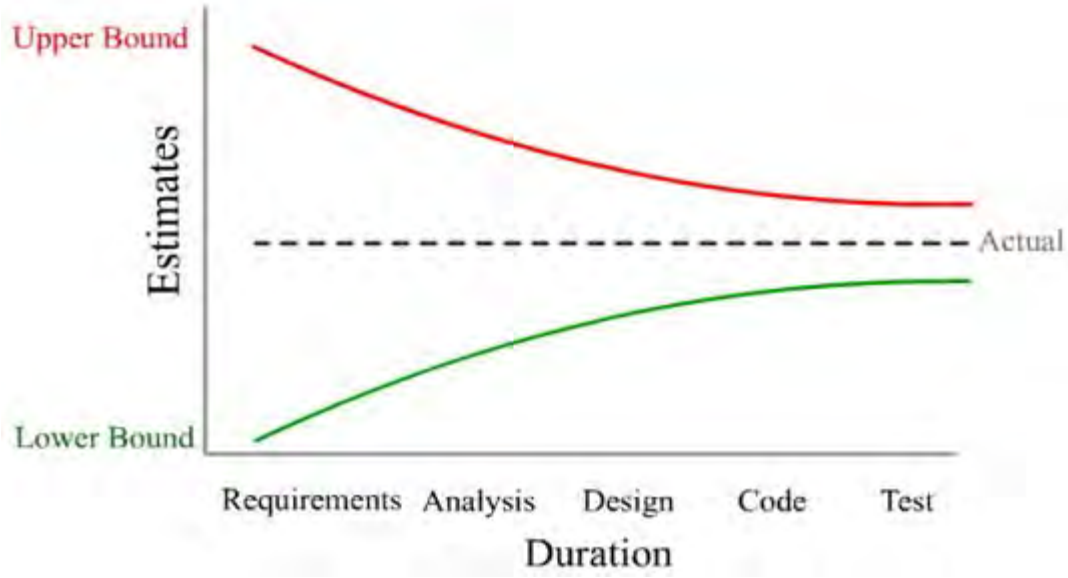

Figure 2: Estimate during the software life cycle

2. Software Estimation Metrics

Several analysts have been designed to discover various techniques and methods that might supply a good method to handle the size estimation. Outputs of those investigations made available a variety of methods, expertly saying known as software Metrics. Today's, quantitative assess is essential in most areas of research as well as in software sciences is the way to provide quantifiable size, which in the case applied correctly creates an effort estimation procedure much simpler and also reliable. Many metrics are suggested however following are the number of which can be typically popular or becoming utilized by various companies. Table 1 illustrates some of these metrics. Table 2 presents a comparison between LOC and FP.

Table 1: Software estimation metrics

Metric	Description
Loc [13]	Number of instructions and data definitions eliminating instructions like notes, blanks, and also continuation lines
Function Point [14]	Measure the functionality of a project. FP estimates are made by determining the indicators of user inputs, user outputs, logical files, inquiries and interfaces
Use case point [15]	The number of use cases and actors, technical complexity and environmental complexity are the main components of UCP
Software Science [16]	Indicates the application of operands and operators , Based on Halstead code length is employed to estimate the volumes or code length

Table 2: Comparing LOC and FP metrics

LOC	FP
Late in the project cycle	Any stage of project
What LOC is	Structured technique
Technology dependent	Technology independent
Sizing is programming style dependent	Sizing is programming style independent

3. Software Estimation Models

Recently there are lots of techniques for effort estimation that are classified into 2 major classes: Algorithmic and Non Algorithmic strategies. From widespread analysis of the publications, we now classify the causes of existing information in 2 vast groups, Algorithmic and Non Algorithmic that is more divided into Linear/non linear models, Discrete models, Multiplicative models, Power Function models, etc. Table 3 consists of a summary of models in their respective classes [17]. Table 4 summarizes a comparison between these two types of estimation models based on ten various viewpoints.

Table 3: Software estimation models

Approach	Category	Model
Algorithmic	Linear / Non Linear	Bailey & Basili
		Farr & Zagorski
		Nelson Model
	Discrete Models	Boeing Model
		Aron Model
		Wolverton (TRW) Model
	Multiplicative Models	Walston & Felix Model
		Doty Model
	Power Function Model	Putnam (SLIM) Model
		Jensen Model
		CoCoMo (Constructive Cost Model)
Non-algorithmic	Expert Judgment	Delphi Technique
		WBS (Work Breakdown Structure)
	Analogy	ABE (Analogy Base Estimation)
		CBR (Case Base Reasoning)
	Learn based	Neural Networks
		Fuzzy
	Soft computing	Optimization Algorithms

Table 4: Algorithmic vs. Non-algorithmic models

Algorithmic	Non-algorithmic
Low flexibility	High flexibility
Model based	Learning based
Much information is required	Some of project attributes are enough
Statistical methods	Various methods
Simple estimation process	May be complicated process
Need to update	Adaptable to new changes
Fast estimation	Often time consuming estimation
No human can interfere the models	Experts can adjust the method
Inaccurate estimations at early stages of project	Accurate estimations at early stages
Usable with especial parameters	Usable with various parameters

4. Software Estimation Approaches

There are lots of methods for breaking-down estimation techniques. The high stage classes are the following:

• Formal estimation type: The quantification stage will depend on specialized techniques, e.g. the utilization of an equation resulting from traditional information.

• Expert estimation: The quantification stage, i.e. the phase in which the estimation is generated depending on judgmental procedures.

• Combination-based estimation: The quantification phase relies upon a judgment or mechanical composition of estimates from distinct origins.

Table 5 presents samples of estimation techniques within every group [18,19].

Table 5: Software estimation approaches

Estimation approach	Category	Example
Parametric	Formal model	SLIM, SEER-SEM, COCOMO
Analogy Based Estimation	Formal model	Weighted Micro Function Points, ANGEL
Group approach	Expert model	Wideband Delphi, Planning poker
Mechanical combination	Combination based	Average of a Work breakdown structure-based and an analogy-
WBS-based approaches	Expert model	Company specific activity templates
Size-based approaches	Formal model	Software Size Unit, Use Case Analysis, Function Point Analysis, Story points-based estimation
Judgmental combination	Combination based	Expert judgment based on estimates from a group estimation and algorithmic model

4.1 Selection of Approach

The proof on variations in estimation precision of various estimation methods and techniques shows that we have not any best approach as well as the appropriate precision of a single method or even model compared to a different would depend highly on the context. This means that various companies get profit from diverse estimation methods. Results, outlined in, that could provide the selection of appropriate method depending upon the required precision of a strategy consist of [20]:

• Expert estimation is in general a minimum of as precise as model-based effort estimation. Particularly, conditions with uncertain interactions and also data of great significance not contained in the model might recommend using expert estimation. This considers that experts with appropriate knowledge can be found.

• Formal estimation types not adapted to a specific organization's private background, is really imprecise. Utilization of own traditional information is accordingly essential if only one cannot be certain that the estimation model's central associations are depending on matching project cases.

• Formal estimation models could be especially helpful in conditions in which the model is designed to the organization's perspective, and/or chances are the experts' estimates will likely be a matter of a good level of wishful ideas.

The most strong identifying , in lots of predicting areas, usually composed of estimates from separate resources, preferable using diverse strategies, are going to on regular enhance the estimation precision. Additionally, other features including simple comprehending or interacting the outputs of a method, simplicity of use of a technique, price of introduction of a strategy can be used in a choice procedure.

5. Conclusions

Software effort estimation is widely considered to be the weakest link in software project management. As it is discussed earlier in our survey that it is really hard to have software cost estimation in early stages as we mentioned that software is something intangible, it is easier to have and estimation bout process in nature however it is much different of intangible products. Software effort estimation is a significant procedure in software life cycle that can never be ignored. To appropriate control of every project, good size has to be used. If you cannot measure it, you cannot manage. During this article we have summarized several estimation techniques/models, besides some methods in non algorithmic section. The article initiated with a brief review of knowing effort estimation and also aspects impacting it. Software estimation metrics were defined in section 2. Algorithmic techniques introduced in section 3 that were dispersed to various classes. While any method introduced in section 3 was not discovered ideal or with no weakness. Almost every method was suggested to resolve the problem once employed in its own environmental border or obviously the algorithms designed as a single environment were not likely capable to be used in another environment. Though, there are numerous possible benefits from utilizing many methods, there is not any solution to find out which methods to apply before processing information.

References

[1] Satyananda, "An Improved Fuzzy Approach for COCOMO's Effort Estimation Using Gaussian Membership Function", Journal of Software, Vol. 4, pp. 452-459, 2009.

[2] Chiu NH, Huang SJ, "The Adjusted Analogy-Based Software Effort Estimation Based on Similarity Distances", Journal of Systems and Software, Vol. 80, No. 4, pp. 628-640, 2007.

[3] Jorgen M., Sjoberg D.I.K, "The Impact of Customer Expectation on Software Development Effort Estimates", International Journal of Project Management, Elsevier, pp. 317-325, 2004.

[4] Jeffery R., Ruhe M., Wieczorek I., "Using Public Domain Metrics to Estimate Software Development Effort", In Proceedings of the 7th International Symposium on Software Metrics, IEEE Computer Society, pp. 16–27, 2001.

[5] Kaczmarek J., Kucharski M., "Size and Effort Estimation for Applications Written in Java", Journal of Information and Software Technology, Vol. 46, No. 9, pp. 589-60, 2004.

[6] Heiat A., "Comparison of Artificial Neural Network and Regression Models for Estimating Software Development Effort", Journal of Information and Software Technology, Vol. 44, No. 15, pp. 911-922, 2002.

[7] Srinivasan K. and Fisher D., "Machine Learning Approaches to Estimating Software Development Effort", IEEE Transactions on Software Engineering, Vol. 21, pp. 126-137, 1995.

[8] Venkatachalam A.R., "Software Cost Estimation Using Artificial Neural Networks", In Proceedings of the International Joint Conference on Neural Networks, 1993.

[9] Selby R.W. and Porter A.A., "Learning from Examples: Generation and Evaluation of Decision Trees for Software Resource Analysis", IEEE Transactions on Software Engineering, Vol. 14, pp. 1743-1757, 1988.

[10] Subramanian G.H., Pendharkar P.C. and Wallace M., "An Empirical Study of the Effect of Complexity, Platform, and Program Type on Software Development Effort of Business Applications", Empirical Software Engineering Journal, Vol. 11, pp. 541-553, 2006.

[11] Huang S.J., Lin C.Y., Chiu N.H., "Fuzzy Decision Tree Approach for Embedding Risk Assessment Information into Software Cost Estimation Model", Journal of Information Science and Engineering, Vol. 22, Num. 2, pp. 297–313, 2006.

[12] Kumar S., Krishna B.A. and Satsangi P.S., "Fuzzy Systems and Neural Networks in Software Engineering Project Management", Journal of Applied Intelligence, Vol. 4, pp. 31-52, 1994.

[13] Fenton N.E. and Pfleeger S.L., "Software Metrics: A Rigorous and Practical Approach", Second Edition, International Thomson Computer Press, 1996.

[14] Pressman R.S., "Software Engineering a practitioner's Approach", McGraw Hill, 2001.

[15] Schneider G. And Winters J.P., "Applying Use Cases: A Practical Guide", 2nd Edition, Addison-Wesley.

[16] Halstead M.H., "Elements of Software Science", Elsevier, 1977.

[17] Abbas S.A., et. al. "Cost Estimation: A Survey of Well-known Historic Cost Estimation Techniques", Journal of Emerging Trends in Computing and Information Sciences, Vol. 3, No. 2, pp. 612-636, 2012.

[18] Zahra S.B. And Nazir M., "A Review of Comparison among Software Estimation Techniques", Journal of Information & Communication Technology, Vol. 5, No. 1, pp. 39-45, 2012.

[19] Lee W.T et. al., "Applying Software Effort Estimation Model based on Work Breakdown Structure", In Proceedings of the Sixth International Conference on Genetic and Evolutionary Computing, pp. 192-195, 2012.

[20] Wen J. et. al., "Systematic Literature Review of Machine Learning Based Software Development Effort Estimation Models", Information and Software Technology Journal, Vol. 54, pp. 41-59, 2012.