



Available Online through
www.ijptonline.com

A REVIEW ON SOFTWARE COST ESTIMATION USING SOFT COMPUTING TECHNIQUES

Srinivasarao.Sabbineni¹, B.V.Anusha², A.Manjusha³, K.Madhavi⁴, E.Akhil⁵

KL University, Department of CSE, India.

Email: srinu1479cse@kluniversity.in

Received on 19-01-2017

Accepted on: 20-03-2017

Abstract

Software Cost Estimation plays a very important role in software development. It involves estimating the effort and cost in terms of money to complete the software development. It is very important when lines of code for the particular project exceeds certain limit, and when the software deployed with too many bugs and uncovered requirements as the project goes incomplete. It also 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. The conclusion is that there is no technique available that is best for all projects in every scenario. Hence a thorough understand and a careful analysis is required for the comparison of all methods in order to come up with best possible method which may lead to the most accurate and realistic estimates. This paper deals with the progress of finding out the best Software Cost Estimation method to give better accuracy on cost estimation.

Keywords: Cost Estimation, Soft Computing Techniques, Fuzzy Logic, Neural Networks, Expert Judgement.

I. Introduction

The problem we discuss and solve here is that there arises many situations in which we are not able to decide a single estimation technique. There are certain parameters to be considered while selecting a technique like no. of parameters to be considered, type of parameters like physical i.e., finance, workers, no. of working hours a day etc., There are many conventional techniques which are very poor in accuracy and estimating the effort. So, we study all the soft computing techniques and try to provide all the parameters in choosing a techniques at the same place so that anyone who studies this paper can have a detailed idea.

1.1. Rationale

Software Cost Estimation plays an important role in every industry. The problem we discuss and solve here is that there arises many situations in which we are not able to decide a single estimation technique. There are certain the

The number of parameters to be considered while selecting a technique like no. of parameters to be considered, type of parameters like physical i.e., finance, workers, no. of working hours a study all the soft computing techniques and try to provide all the parameters in choosing a techniques at the same place so that anyone who studies this paper can have a detailed idea. We read many circumstances in which the team failed to deliver the software in time due to improper estimation. While estimating software effort, the main issue to be considered is avoiding errors, thorough checking of software at all levels like requirement, analysis, and design. Improper checking leads to increased wastage of money, labour. By studying all the existing techniques and published papers, we are able to say that when expert estimation is combined with planning poker technique which mainly focuses on group discussion achieves knowledge sharing. When group estimate is compared with that of individual estimate, the yield further increases quality. Like the above, we can even merge several other techniques to find accurate solutions to the present day problems. We study all the latest soft computing techniques so that we may be able to propose legitimate and consistent solution by integrating some of them. We study different existing soft computing techniques, their advantages and disadvantages and ultimately try to propose a technique which suits every project.

2. Statement of Purpose

The main purpose of this term paper is it deals with estimating software effort accurately which plays a prominent role in the development of any software. Proper Software. Cost estimation helps a team in completing the project in time. So choosing the best model is necessary. Even though there are many techniques already existing, there is not even a single one which suits all. So, we study all the techniques and try to integrate then if possible in order to find a better solution.

3. Important Research Questions:

Before getting in to any project, we must have a clear idea of the product to be developed, how to develop it and so on. An idea on different estimating techniques is necessary. Let us see some of those questions that this paper would help in answering.

3.1. On What Basis, We Say That The Software Cost Estimation Is A Difficult Task?

The main reasons would be: i) Lack of information about existing software projects.

ii) Estimation is always done in a hurried manner without any reward for the work done.

iii) The estimators are forced to complete the work that they even do not understand.

iv) Clear and reliable requirements at the early stage are difficult to be done.

v) The factors that affect the cost are called cost drivers like size, complexity and the user involvement. But these cost drivers are difficult to recognize.

vi) Changes made and adding new facilities will disturb the estimated cost.

vii) The estimator himself get in to a conclusion about the cost without thinking of the caliber and the productivity of the staff.

3.2. What Would Be The Reasons For Exceeding Of Budget And Planned Durations?

Improper monitoring of costs at each level leads to budget run out. If the manager takes care of it by inspecting at every level, it wouldn't happen.

The planned durations may be disturbed as the employees appointed on a particular task falls sick or on a leave for some personal reasons. Not only this, after getting in to the project, we may fall in dilemma that how many staff would be required for a task. So, in this state we appoint someone who is not efficient enough. It also results in the delay. Less team spirit.

3.3. What Will Be The Prerequisites For Estimating Cost?

i) Clear cut idea of the requirement specified by the client.

ii) An overview of software cost estimation models.

i) Productive Staff.

This would be an important question, so lets go in depth. This solution deals with words like WHO, WHAT, WHEN.

i) Who are the end users?

ii) What are the specifications?

iii) When should the product be submitted?

4. Basic Terms

There are some terms which should be clearly understood. So. let us see in to those terms once.

4.1. Soft Computing

Soft Computing means the usage of inexact solutions to solve computationally hard tasks such as the solution of NP-Complete problems, for which there is no existing algorithm that can compute an exact solution. But these are unpredictable, uncertain and ranges between 0 and 1. Soft computing differs from conventional(hard) computing in that, it is tolerant of imprecision, uncertainty, partially true. In effect, the role model for soft computing is the human mind.

4.2. Software Effort

In software engineering, effort can be used to denote the measure of use of workforce and the total time taken by the members of a development team to perform a given task. It is usually expressed in units such as man-day, man-month. This value would be important as it becomes a basis for estimating other values relevant for software projects, like total cost or time required to produce a software product.

4.3. Software Effort Estimation

Effort Estimation means the process of predicting the amount of effort which is expressed in terms of person-hours or money and required to develop and maintaining the software based on incomplete, uncertain and noisy input. Effort estimates may be used as input to project plans. There are many soft computing techniques like regression analysis, neural networks, fuzzy logic, Analogy Cost Estimation, Genetic Algorithm, COCOMO Model, COCOMO II Model etc.,

5. Differences between soft computing and conventional techniques.

Table 1: Comparing Soft Computing and Conventional Techniques.

S.No	Conventional Techniques	Soft Computing Techniques
1.	Needs a clearly stated analytical model.	Tolerant of imprecision.
2.	Requires a lot of time for computation.	Can solve in reasonably less time.
3.	Does not solve real world problems.	Solves Real World problems.
4.	Requires full truth.	Partial truth is sufficient.
5.	Accurate and Precise.	Imprecise.
6.	High Cost	Low Cost.

Unique Features of Soft Computing:

- i) Fast and intelligent in computation. Contains human like expertise in a domain.
- ii) Able to explain their decisions.
- iii) Adaptive to changing environment.

6. Factors Affecting Software cost Estimation

We will come to know that the effort estimation depends on the outputs of the following queries.

They are:

- i) What is the effort that is to be taken to finish the activity?
- ii) What is the approximate time required to complete each sub-activity?
- iii) What will be the total cost of each activity?

Usually, the cost estimation and scheduling of project activities are done simultaneously. The advantage is that we can lessen the price when we think that our project exceeds the estimated cost, so that it would be easier to make any changes. If we go for estimating at deployment phase, it would be a burden to us in making those changes.

Truly saying, the cost of development depends on the cost of efforts included in each phase. So, the effort should be estimated in each activity.

Primarily, the cost estimation depends following factors.

- i) The costs of hardwares and softwares included.
- ii) Cost of Maintenance.
- iii) Costs of efforts like paying the Software engineers involved.
- iv) Cost involved in studying the project.

For almost all the projects, the most important cost is the cost of effort. We can use computers which are capable of developing a software. But the thing is that when a project is developed at different sites, the travel cost dominates the effort cost. Moreover, using electronic communications like emails reduce the purpose of travelling required. This also makes the usage of time more efficient. Cost estimation does not only depend on the salaries given to the staff, but also the no. of heads worked on the project which is done by dividing the total cost to run an organization with the no.of staff worked on the project. So, the following are also inclusive in the estimation. Let us see them.

- i) Cost of providing an effective working environment.
- ii) Cost of supporting staff who accounts, administers and other technicians.
- iv) Cost to communicate with the end users or the staff itself.
- iv) To pay for other referential activities like libraries and other recreational activities.
- v) Cost to provide security to the project details and other employee benefits like housing loan etc.,

6.1. Overhead Factor

Usually, the overhead factor is the dominant one when compared to the project cost estimation. But, it always depends on the organisation's size and its accompanied tasks. In short, We can say that the overhead is double the amount that is paid to a software engineer per year.

When a project is under development, it's each activity cost should be regularly updated so that the project manager can deal with it. When a project's cost estimation is done by developers, they have to deal in giving high quality with low cost. But when the cost is bid by the client, We have to act according to the budget. So, we appoint senior

managers who can act according to the situations with other helping staff. Now let us see some of the factors that affects the pricing of the software.

Table 2: Factors affecting the Pricing.

Factors	Explanation
Market opportunity	When a new company bids for a project with less profit, it means that it hopes to get huge profits later and experience.
Uncertainty in cost estimation	When it is uncertain in estimating, it bids with huge profit.
Terms of contract	When a customer asks to allow the developer to use the code in other projects, it costs less.
Financial health	When a developer is financially in critical position, he may agree to work for less salary.
Changing Requirements	When the requirements are changed by the customer after the deployment, it costs high.

Software cost depends on the software productivity. The most frequently used metric is that the no. of lines that a programmer codes in a month This can be denoted by:

$$(\text{Loc}/\text{Month}) = (\text{No. of lines of code} / \text{No. of months taken}).$$

When we use the measure of functionality of the code as a metric instead of code size, we may get out of the impression that the functionality is independent of the language used.

7. Techniques of Estimation

It won't be a simple task to estimate the cost at earlier stages we have to make initial decisions according to the users requirements.

The software should be able to run on all the systems and use the latest technology.

We will come to know the skills of the people involved in the later phases.

To estimate the cost, it requires the experience of senior managers even though they may not be aware of the latest technologies. The new technologies may be:

- i) The usage of web services.
- ii) Distributed systems rather than single centred system.
- iii) Using Database Systems.

Now let us see each of them in brief.

7.1. Regression Analysis:

It is one of the most commonly used techniques to uncover the hidden software bugs in existing functional and non-functional areas of the system.

The main advantage is that it helps us in understanding whether a change made to one part of the system affects the other parts. It is used to estimate the efficiency of the system by checking it frequently.

7.2. Case Based Reasoning:

It is a type of machine learning that uses past experiences in order to find solutions to current problems. We commonly do not try to think of a solution, we show interest in using already existing solutions. This model doesn't have any dedicated world model logic, rather deals with past ones. It consists of 4 R's. They are:

- i) Retrieve: It means finding the most similar cases to the current ones.
- ii) Reuse: Adopting the techniques used to solve the past problem.
- iii) Revise: Check whether the old ones solve the new ones.
- iv) Retain: Retaining the parts of the current experience in the case base for problem solving.

7.3. Bayesian Statistics

When new data becomes available, we update probability distribution, in that way it supports incorporation of new data. It is a subset of the field of statistics in which the evidence about the true state of the world is expressed in degrees of belief or more precisely, Bayesian statistics. Such an interpretation is only one of a number of interpretations and there are also some other statistical techniques that are not based on "degrees of belief". The key idea of this is that probability is opinion ordered, and that inference from data is nothing other than the revising of such opinion by using relevant new information.

7.4. Genetic Algorithm

It is a search heuristic which minimizes the natural evolution. It is normally used to generate optimal solutions for search problems. It normally uses inheritance, mutation, selection and cross over.

Let us see them in brief.

7.4.1. Inheritance:

It means acquiring the properties from the parents without the problem of recreating them.

7.4.2. Initialisation:

In this phase, we try to find all the possible solution for the problem which is called as problem space. Usually the size of the problem space depends on the complexity of the problem.

7.4.3. Selection:

In the selection phase, we find the solution which gives the optimised result among the problem space. This is done using techniques like Boltzman Selection, Roulette Wheel Selection, Tournament Selection and Rank and Steady State selection.

7.4.4 Cross Over: It is a operator to recombine where in two individuals are crossed to form an offspring (new solutions) in order to acquire special properties. It can be Single Site, Two Point, Multi point Cross over, Matrix Cross over.

Advantages:

- i)Applicable even when little information is known.
- ii)Better way to find solutions to the complex problems quickly.

Disadvantages:

- i)Poor Performance.

7.5. Cocomo Model

It is derived by gathering data from number of projects. This is analysed to get a formulae that would best fit for all kinds of projects. It links project size and other factors.

This model contains relevant information with number of parameters and their ranges.

The Constructive Cost Model has 3 variants. They are :

- i) Basic Model
- ii) Intermediate Model
- ii) Detailed Model.

The selection of the model depends on the type of the software required. But the basic cocomo model computes the effort E as a function of program size.

$$E=a*(size)^b$$

Where a, b vary according to the complexity of the software. But the intermediate model describes effort in terms program size and set of cost drivers for effort multipliers. $E=a*(size)^b*EAF(2)$ EAF means Effort Adjustment Factor which in turn is calculated using 15 cost drivers. But the intermediate model describes effort in terms program size and set of cost drivers for effort multipliers. $E=a*(size)^b*EAF(2)$ EAF means Effort Adjustment Factor which in turn is calculated using 15 cost drivers. Where as in detailed COCOMO model, the requirements gathering, planning and product design, code design, unit testing and integrated testing are done.

7.6. Cocomo II Model

The COCOMO model is updated as COCOMO II model which consists of Application Composition model, Early Design and Post Architecture model.

Early Design Model is used to evaluate software architectures where unadjusted function is used for sizing. Post Architecture Model is used during the actual development of a product and includes cost drivers. The post architecture model is the important one. After designing the architecture of a system, a better estimate can be made. But it also uses the basic formula. The most detailed one among the cocomo models is the post architecture model. As the cocomo model depends on the size, the following parameters are to be observed.

- i)An overview on number of lines of code to be developed.
- ii)The no.of lines to be modified due to the requirement changes. The factors used in adjusting the initial estimates fall under 3 categories.
 - i)Concerned with requirements of the required project.
 - ii)Computer attributes which means the conditions that are imposed on the software.
 - iii)Personnel attributes means those that consider the experiences or the abilities of the staff.

7.7. Artificial Neural Networks

It is usually used as a substitute for regression technique and also used to estimate cost in financial services, bio-medical applications, time-series prediction, text mining and decision making.

For cost estimation, different input signals are weighed and combined in to a final cost model. By feeding every part with a cost, it is trained to estimate the total cost. These are nets of processing elements that can understand the mapping between input and output data. neuron calculates the weights of the inputs and generates an output if it reaches the threshold value. Then this output becomes input to remaining neurons in the network. The process continues until one or more outputs are generated. The network checks all the possible consequent layers to the output layer. Every neuron calculates the non linear function of inputs and pass it along with its output.

But neural networks is limited in several aspects, it is rarely used. It is a black box approach, so difficult to understand the internal functioning. So the effort prediction is complex when compared to other models. Generally, it is capable of dealing with classification problems but we need generalization capability.

7.8. Fuzzy Logic

When there exists uncertainty and imprecision while estimating software cost, we use fuzzy logic. It doesn't has any limitations and used to solve complex problems. It has 4 four main components. They are:

- i) Fuzzifier: Converts the crisp input in to a fuzzy set and uses graphical functions to describe a solution.
- ii) Fuzzy rule: Uses If-Then rule.

iii) Fuzzy Inference Engine: It performs operations like Aggregation and Composition using If-Then rule.

iv) Defuzzification: Converting the fuzzy output to crisp output.

7.9. Estimation by Analogy

It is a technique which is used to estimate cost based on historical data of a sub system. To this, We use only the systems which are similar to the current one. The cost of the present system is found by making some differences to the previous systems.

These adjustments can be regarding size, cost, performance, technology etc.,

We still improve the accuracy by adding some heuristics.

7.9.1. Heuristics

It is also a process of gaining knowledge or some desired result by some brilliant guess work rather than using some pre-defined formulae.

Chi-Squared Analysis is also very useful in determining the significance of statistical association rules. It can be computed from the values of confidence and support.

7.9.2. Principal Component Analysis

It is a which depends on statistics and uses an orthogonal transformation to convert a set of observations of possibly correlated variables in to a set of values of linearly uncorrelated variables called principal components. But, this may not improve the accuracy of all projects, it works only for specific projects. The steps used in this technique would be:

- i) Checking and thoroughly studying the proposed project.
- ii) Picking the most synonymous one out of many similar ones.
- iii) Applying that technique to derive an estimate.

Advantages:

- i) The estimation depends on the actual project data.
- ii) The experience and talent of estimator is used.
- iii) The difference between the reference and proposed project can be easily understood.

Disadvantages:

It would be difficult in calculating the amount of similarity. It depends on the number of inputs, quality of output and number of references made to get in to a conclusion.

7.10. Function Point Analysis

Function Points is not a concept which is new to us but got greater acceptance as a software size measure recently. These usually measure size based on the functionalities required and accepted by the user. These also represent logical size. The more the complex the function is, the more they contribute a higher to the logical size.

- i. Data Functions perform the logical grouping of data end users to complete their jobs.
- ii. The complexity always depends on the no.of data elements and logical sub groupings.
- iii. Internal data is always maintained by the application and the external data is referenced by the application.
- iv. Transactional Functions includes the processes and actions that the users use to change and manage the data while performing their job.

Like inputs : add, edit, delete etc.,

Outputs : reports

Inquiries: Search, Retrieve etc.,

It always provide a consistent, documentable and repeatable measurement methodology. These also act as a means to communicate between the end user and the developer. End Users would not specify the size of the application ,they say only the requirements needed.

7.10.3. Advantages:

- i) It will be viable from early requirements stage and applicable to full life cycle analysis.
- ii) Platform independency.
- iii) Advantageous over lines of code.
- iv) Repeatable and Documentable.

7.10.4..Disadvantages:

- i) Accurate counting requires in-depth knowledge of standards.
- ii) Requires man power.
- iv) Some variations also exist.

Usual typical software estimation methodologies can serve as function points as the key size input.

Manual: Calculations which can be done easily on a sheet or by using a calculator.

Automatic: Models or tools which are driven by more complex internal algorithms and depends on project history data and the relationships between the existing cost drivers.

i. Cost, Schedule Re-estimation are important regarding effective structured change control process.

ii. Requirements of a project keeps changing frequently.

Before incorporating any change, its impact and cost should also be evaluated.

i. According to the impact we obtained on calculating, we decide whether the change is to be incorporated.

ii. Function Points compare the actual performance with the time phased plan.

iii. These are linked to requirements directly, so provides a traceable methodology for re-sizing the software.

iv. These help in identifying gaps in requirement analysis to avoid early introduction of defects.

v. It also offers a chance to cross verify the completeness of the requirements and to understand how they are developed.

7.11. Expert Judgement Method

It means consulting experts to use their experience or ability to arrive at a cost estimate. Delphi would be the best word to describe this. The procedure is:

i) Project head would call all the experts and give each expert with an estimation form to fill.

ii) Calls for a group meeting where the experts discuss certain issues with the project head.

iv) The experts fill out the estimation form.

v) The project head studies each form and prepare a summary of where their opinions are varying to a large extent and distribute to the experts.

vi) Then again calls for a meeting and discuss points where all their opinions vary.

vii) The experts again fill the forms and repeat the above steps until the requires solution is obtained.

Advantages:

i) The experts can find the differences between the previously dealt projects with the requirements of the latest project.

ii) They can also estimate that would be caused on the previous projects when they use the latest technologies.

Disadvantages:

i) It becomes hard in documenting the parameters used by the experts.

ii) The experts may have broken the group consensus.

iii) This method uses the already used methods so that it may be that effective when compared with the previous results.

7.12. Top Down Approach: This can be also called as Macro Model. In this method, we can estimate the overall cost using the required properties. Then we partition the project in to several small components.

7.12.1. Advantages: i)It will never miss even the system level activities like documenting, combining etc., which are neglected in other approaches.

ii)It requires minimum details about the project, so that it can be faster and easier.

7.12.2 Disadvantages: i) Frequently misses the low-level problems which increases the cost of the project.

ii)It doesn't provide the details of why a particular decision has been made.

7.13. Botttom –Up Approach:

This method deals with the cost of individual software component and then integrates the results obtained to get into overall estimation. It focusses to construct the estimate from the estimate from the information collected about small components. COCOMO Model uses this approach.

7.13.1. Advantages: i) More stable because the errors in estimation may get a chance to level.

ii) It also gives even the freedom of using the traditional methods if required.

7.13.2. Disadvantages: i) It fails in noticing the system level costs like combining , assurance of quality etc.,

ii) It would not be accurate enough as the basic information won't be viable in the early stages.

iii) Time consuming.

8. Conclusion: After studying all the soft computing techniques and conventional techniques, we understood that soft computing techniques are better at accuracy, certainty. But soft computing techniques also may not be good at giving accurate results for every project. On combining 2 or more soft computing techniques, we may be able to get better results. We study different existing soft computing techniques, their advantages and disadvantages .In feature we try to propose a new technique which suits every project for cost estimation in advance.

9. References

1. C.J.Burgess, M. Lefley, (2005), "Can Genetic Programming improve Software Effort Estimation? A Comparative Evaluation", Machine Learning Application. In Software Engineering: Series on Software Engineering and Knowledge Engineering, pp.95–105.
2. Finnie, G. R., G.E. Wittig and J-M.Desharnais, (1997), "A Comparison on Software Effort Estimation Techniques Using Function Points with Neural Networks, Case-Based Reasoning and Regression Models", Journal of Systems and Software, Vol. 39, pp.281-289.

3. G.; Finnie, G., "Using Artificial Neural Networks and Function Points to Estimate 4GLSoftware Development Effort", J. Information Systems, vol.1, no.2, pp.87-94, 1994.
4. X. Huang, D. Ho, J. Ren, and L.F. Capretz, "A soft computing framework for software effort estimation," Soft Computing, vol. 10, no.2, pp. 170–177, 2006.
5. Idri, A. Abrian, and L. Kjiri, "COCOMO Cost Model using Fuzzy Logic," International Conference on Fuzzy Theory & Technology Atlantic, New Jersey ,2000.
6. J. Lu, B. Xu, and H. Yang, "A Classification Method of Fuzzy Association Rules," IEEE International Workshop on intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 2003.
7. H. Park and S. Baek, "An empirical validation of a neural network model for software effort estimation," Expert Systems with Applications, vol. 35, no. 3, pp. 929–937, 2008.
8. Y. Shan, R.I. McKay, C.J. Lokan, and D.L. Essam, "Software project effort estimation using genetic programming," IEEE International Conference on , Circuits and Systems and West Sino Expositions, vol. 2, pp. 1108–1112, 2002.
9. Schofield C., 1997 "Estimating software project effort using analogies", IEEE Transactions on Software Engineering 23 (12) 736–743.
10. Srinivasarao.Sabbineni, Kurra. Rajasekhara Rao "Assessment of Architecture-based Software System Reliability Allocation on components using a Dynamic programming". IJCA, 2013, 73(19) :41-45
11. Srinivasa Rao. Sabbineni, Rajasekhara Rao.Kurra "Estimation of Reliability On Components using a Dynamic Programming". IJCSI, 2013, 10(3) :78-81.
12. Srinivasa rao. Sabbineni, Dr.Rajasekhaara rao.K, Kiran. V.D"Analysis of software reliability and cost assessment relationship for architecture-based software" GJCAT,2011, 1(3) :403-406
13. Srinivasa rao. Sabbineni, Dr.Kurra. Rajasekhara Rao "A Comparitive Study On Reliability And Cost Estimation For Architecture-Based Software" IJAER,2015, ISSN 0973-4562 Volume 10, Number 8 (2015) pp. 19729-19736
14. Srinivasa rao. Sabbineni, Dr.Kurra. Rajasekhara Rao "An Accurate allocation of Reliability and cost estimation for Architecture-based Software using DRMPA" IJARCSSE,2016, ISSN 2277 128X Volume 6, Number 11 (2016) pp. 43-50.