Framework Of Software Cost Estimation By Using Object Orientated Design Approach

Farooq Azam, Salman Qadri, Shabir Ahmad, Kamran Khan, Abu Buker Siddique, Bilal Ehsan

Abstract: Estimating software development cost is vital issue because cost is the major factor of any project. If anyone want the assessment of software estimate cost its seemed to much tough job. Cost estimator used different techniques and methods that not too much efficient. Estimating the cost on the basis of object oriented design approach too much effective. After drawing the basic architecture basis on Classes, Methods, Objects, entities, and their relationship may gives the better result to estimate the cost of software. This paper discusses the strengths and weaknesses of existing size estimation techniques, considers the nature of software cost estimation, and presents a software cost estimation model which has the potential for providing more accurate cost estimates than existing methods. The proposed method takes advantage of a characteristic of object-oriented systems, the natural correspondence between specification and implementation, in order to enable users to come up with better cost estimates at early stages of the software development cycle.

Index Terms : Software Cost Estimation, Object Oriented Approach, Software Development, Software Size

1. INTRODUCTION

Cost estimation of computer system projects play critical role in the development of all projects. If estimated cost less than the actual cost then you may face number of issue effect the quality, purposed time to complete, pressure on developer to give overtime. A general issue in programming cost estimation is, that verifiable expense information regularly holds a inclination, with the goal that this information cannot straightforwardly be utilized for advancement of expense estimation devices, individually for the adjustment of such instruments and looking at estimations against its true (recorded) costs. Accurate cost and scheduling estimations provide highly valuable aid in a number of management decisions, budget and personnel allocations, and in supporting reliable bids for contract competition. Cost estimation models today available for the software engineering practitioner include: the Walston-Felix model [1], the Doty model [2], the Putnam model 1251, the RCA PRICE S model [4], the Bailey-Basili model [3], the COCOMO (Constructive Cost Model) model [3], the SOFTCOST model 1341, and, recently, the Jensen model 141. In [5], a complete description of the COCOMO model together with a very detailed approach for its utilization on the daily life of an organization can be found. The Secure access of software in the organization is observed by the MLA technique by Ahmad Shabir and Bilal Ehsan [6].

1.1 The problem of object-oriented software estimation

Even though estimation models have some limitations that managers need to be aware of [3], they may be viewed as valuable tools in the software engineering process. A common point concerning the above mentioned models is that they base their effort and scheduling predictions on the estimated size of the software project at hand, in terms of number of lines of code (LOC), or thousands of lines of code (KLOC). Generally the effort estimation is based upon an equation similar to

\[ E = A + B \times (KLOC)^C \]

where \( E \) stands for estimated effort (usually in many months), \( A \), \( B \), and \( C \) are constants, and KLOC is the expected number of thousands of lines of code in the final system. From the above equation it is easy to see that a given percentage error in the size (KLOC) may cause an even larger percentage error in the estimated effort. For instance, in COCOMO a 50% error in the size estimate will roughly result in a 63% error in the effort estimate.

1.2 Software Cost Estimation Difficulties

Evaluating programming improvement sets back the finances with correctness is extremely troublesome regular approach for enhancing programming cost gauges is to utilize exact models. The extra objective of having the capacity to anticipate the expenses and calendar at the start of the venture can end up being all the more testing. "Unanticipated forecast of finishing time is completely vital for fitting development arranging and abhorrence of the Nevertheless, utilizing the whole suite of accessible programming cost estimation displays, scientists uncover that there is no confirmation that programming models are adequate at assessing ventures at an early phase of framework improvement off base.

Best Practices of Cost Estimation

The accompanying is particular rundown of programming cost above consolidated with class. Worldwide Social order of Parametric Experts, classified it into four classes.

- Infrastructure for establish cost estimation
- Performance of the Estimate
- Estimation of the documents
- Performance Monitoring

2. LITERATURE REVIEW

Zulkelili Mansor et al papers "Software Cost Estimation" [7] In conclusion, scrutinize in expense estimation prepare has been done for a considerable length of time with immense number of investigates need to investigate the different angles in the estimation transform. This study has taken a
gander at the triumph variables of accepted and dexterous cost estimation prepare in programming improvement extend. Hareton Leung , Zhang Fan [8]. Number of cost estimation models estimates the cost of software with accuracy but there are number of factors that complex this process. Rapid change in technology, development tools, languages that not ensure the 100% accuracy. F J Heemstra paper “Software Cost Estimation” [9], find out the level of ambiguity during the cost estimation process so no matter ambiguous and create problem in development process. When you estimation cost collect date of all type without hesitating try not missed any type of data Use the number of technique not relay on one or two technique rather than this use mere then one technique Commitment of all type of need in detail which required for the software cost estimation. Prof. M. GlinzArun Mukhlja Seminar “Sources of Error in Software Cost Estimation” [10], One precondition to get exact cost estimations is to get precise require information from previous programming undertakings, which could be utilized to confirm the correctness of an estimation or an estimation tool/method and to balance the estimation database. There are numerous elements which impact the expenses of programming ventures. However the use of a programming cost estimation apparatus is no surety for an exceptional estimation, the estimator has still to have much experience and information about previous undertakings to get great results. Prof. Dr. Abdul Hamid M Ragab, Dr. Abdulfattah Suliman Mashat, Dr. Ibrahim Elbedewi paper “ Automated System for Software Project Cost Estimation Based on Ontology Engineering” [11], The proposed framework is executed of three fundamental segments. The first segment verifies the life cycle which incorporate the cosmology building situation. The second part incorporates detail of the measure of cosmology to be constructing, communicated in many ontological primitives (notions, relations, sayings, and examples). The third segment incorporates detail of expense drivers rating for a task, relating to the data accessible. The framework is actualized utilizing Microsoft visual studio Asp.net C# that could be run effortlessly on Pc. Jennifer Leotta paper “Software Cost Estimating Relationships”[12]. So as to acquire exact cost appraisals for programming improvement programs, it is best if an association gathers information in regards to variables that are best at evaluating the expense particularly, program estimate, crest staff, and programming necessities; which appear to be the best variables to verify the project. By gathering information and administering a company/program particular database, a basic and viable relationship might be utilized to produce a Rom cost gauge for programming advancement fetches. MICHAEL A. ROSS paper “Joining Effort and Duration in a Probabilistic Method for Predicting Software Cost and Schedule” [14], The Software Estimating Framework is information driven programming evaluating procedure that characteristics a summed up cost and duration estimating relationship arrangement of two comparisons. The strategy is totally open as confirm by the way that the paper holds sufficient detail with cases to allow sensibly simple usage and deployment.es reporting that fulfills all stakeholder prerequisites. Kirsten Ribu “Estimating Object-Oriented Software Projects with Use Cases” (2011) provided that there are no organization guidelines for the written work of utilization cases, different groups might compose utilize instances of exceptionally different structure and structure for distinctive subsystems of a bigger framework, as was seen in case study. Different methodologies are acknowledging the measure of reuse, or the amount of interfaces and database passages on which the utilization case touches as an estimation of intricacy.

3. METHODS AND TECHNIQUES
Main two types of the cost estimation methods: algorithmic and non-algorithmic. Non-algorithmic methods based on general idea while algorithmic method purely base on mathematical and statistics formulas

3.1 Non-Algorithmic Methods
Expert Judgment: In the accompanying segments, diverse master based estimation methodologies are introduced. Rather than parametric models, where the estimation step is made mechanically and dependent upon a scientific equation, master built estimation depends with respect to a human estimator for the estimation step. Rather than master based estimation, the at one time displayed parametric models are simpler to investigate, since scientific models are for the most part less complex to study than strategies including human conduct. Under particular conditions a master based estimation is more relevant than a parametric model. Case in point, it may be difficult to gauge ventures with high specialized questionable matter utilizing a parametric model.

Price to Win: This technique objective to win the project by compromising on cost. According to customer affordability ignore the quality of the software .This technique is not good because software quality compromised in this way , also delay to delivery date and overburden the employee.

3.2 Algorithmic Methods
Algorithmic method based on mathematical formula to formulate the cost estimate operator used with variables for calculation. There variable are the cost factors elements.

Ex: Effect={a1,a2,a3,.................an}

Where F(a1,a2,a3,.................an) are the cost factors given algorithmic have two aspects the cost factors and the function F. First we work out on the cost factors then it function F in point of view is empirical and analytical.

3.2.1 Software Size
Size is a standout amongst the most paramount qualities of a programming item. Marker of programming cost and time it is likewise a base unit to infer different measurements for programming extend estimations, for example profit and
surrender thickness. This segment portrays the most prevalent estimating measurements and methods that have been proposed and connected in practice. These strategies could be ordered into code-based estimating measurements what's more utilitarian size estimations.

3.3.2 Line of code
Total number of lines of the “Pure Source Code” known as LOC. Source code line depend on the programming languages it vary according to the language and its estimated complete when software complete. It’s tough to estimate using this technique. Expert judgment technique used to estimate the line of code. The name of this technique is called PERT it base on three assumed parameters by experts. \( S_L = \)Lowest estimate of code of lines. \( S_H = \) highest estimate of code of lines. \( S_M = \) Medium estimate of code of lines. 

\[
S = \frac{S_L + S_H + 4S_M}{6}
\]

3.4 Framework of the Software Cost Estimation

![Figure 3.1 Framework of The Software Cost Estimation](image)

4. EVALUATION AND FINDING
The programming organization required a technique or apparatus for registering correct appraises at an unanticipated advancement organize. The necessities for such a strategy on the other hand device were accumulated from meetings with the undertaking director of project

**Delphi Survey Results**
Master judgment estimation, as the name suggests, is an estimation procedure that depends on the specialists to handle venture gauges dependent upon their experience as restricted to utilizing formal estimation techniques. Master judgment estimation is suitable in the situation where data is not sufficient or an abnormal amount of questionable matter exists in the venture being assessed. Boehm 2000b [14], of numerous master judgment systems presented, Wideband Delphi has been connected effectively in confirming introductory rating values for the COCOMO-like models, for example COCOMO II. In this study, the Delphi practice was likewise utilized to arrive at master accord with respect to introductory rating scales of the support exertion demonstrate. The outcomes are treated as prior-learning to be utilized as a part of the Bayesian and other alignment methods.

**Software Maintainability and Supportability**
Programming support and underpin is a huge take driver over the sum life of the project. It is basic that programming designers outline programming to be solid, justifiable and modifiable. Enhancing the starting outline will surely diminish the pattern where support and underpin expenses are 40-80 percent of the sum programming life

![Figure 4.1 Cost for Data Processing Environment](image)

Cycle cost. Figure 1 speaks to current support and underpin information for information preparing programs. In both cases the support fetches for these projects might be huge.

5. CONCLUSIONS
A technique for software size estimation has been proposed. The basis for estimates, when using this method, is the available knowledge of the considered system. In order to capture and represent this knowledge, an object oriented functional model has been adopted. This functional model provides for a disciplined methodology for decomposing system complexity. This methodology is the key in the process of detailing the functionality of the system in order to enable estimators to achieve more reliable estimates. In this theory we investigated the effect of value prerequisites, specifically security and convenience on assessments made utilizing and master estimation. The objective of this proposal was to explore the favorable circumstances and confinements of formal estimation systems vs. master estimation with respect to quality necessities. In a survey we talked over the studies from the literary works on the execution and suitability of parametric models and master estimation for certain situations. We introduced the conclusion of a arrangement of meetings with expense estimation specialists and shed light on the
provision of diverse expense. At the middle of this research, we directed two examinations: In the predominant analysis we examined the effect of security and ease of use prerequisites on gauges made by understudies in the part of master estimators contrasted with gauges made utilizing the said models. In the second examination we examined the effect of convenience necessities on the gauges made utilizing these methods. The members of the second examination were understudies, as well. They chose the information parameters for the estimation of a fundamental framework and for the estimation of the same essential framework, however with extra convenience prerequisites. We then dissected the effect of the convenience prerequisites on the capacity focuses check (measure information) and the gauges for exertion, calendar and cost made utilizing.

6. FUTURE WORK
There are number of bearings for future work that are worth investigating. These bearings include further balancing the model, amplifying it to different sorts of support, to iterative and deft programming advancement techniques, and enhancing the compelled relapse approaches. This work has demonstrated execution change in adjusting the model to program, basically benefiting from normal characteristics of all discharges in the same program, for example provision dominion, customizing dialect, and working stage. Building the dominion particular, dialect particular, or stage particular demonstrate by balancing the bland model utilizing information sets of the same feature(s) would conceivably enhance the estimation precision. The extent of this work was restricted to useful improvement and issue rectification sorts of programming support. This limit was infringed, in part, by the information set gathered for adjustment and approval. Future work might be amplifying the model to uphold different sorts of programming upkeep, for example reengineering, switch designing, dialect and information relocation, execution change, program flawlessness, documentation, preparing, and other practical backing exercises.

References


