

Cost Estimation Tool for Commercial Software Development Industries

Manisha Arora^{#1}, Richa Arya^{*2}, Dinesh Tagra^{#3}, Anil Saroliya^{#4}, Varun Sharma^{#5}

^{#1}ASET, Amity University Rajasthan, Jaipur, India

^{*2}ASET, Amity University Rajasthan, Jaipur, India

^{#3} Dalhousie University, Nova Scotia, Halifax, Canada

^{#4}ASET, Amity University Rajasthan, Jaipur, India

^{#5}ASET, Amity University Rajasthan, Jaipur, India

ABSTRACT: As cost associated to the software development is increasing continuously, there is a need to direct attention at better understanding of development process and calibrating previous models and methods making them applicable to existing environment such as proposing hybrid tools using the techniques based on existing models. The main aim of this paper is to improve the cost estimation accuracy of applications at all the phases of a software development lifecycle by calibrating COCOMO using a function point as a size measure instead of SLOC being used in COCOMO model. Here we have discussed the working of proposed software estimation tool that is a hybrid implementation of various software estimation and measurement techniques helping an organization to determine metrics such as Effort, Time and Cost, essential for improving turnaround time. This tool also works towards utilizing these metrics for project planning, scheduling and tracking.

Keywords- software cost estimation; software estimation tool; metrics; project planning; insert (key words)

I. INTRODUCTION

Estimating the cost of the software is still one of the most challenging tasks in Software Engineering. Estimation of software projects is one of the management activities and despite its complexity it is sometimes performed only by the experts themselves. Software cost estimation is more difficult than cost estimation in other industries, as software development is human-based activity with uncertainties that leads to difficulty in cost estimation mainly during early stages of project.

The purpose of software cost estimation is to:

- Determine the resources needed for the project, and their utilization.
- Determine the uncertainty and risk involved.

The estimation at various stages is error prone or difficult to perform during early stages. Possible approach is to measure the size of the project or estimate its cost based on judgments.

But, according to research statistics difficulty faced by various researchers was when the complexity and size of the software increases [5].

The main problem which came up was the dependency of organization on the past experience or historical data for estimating the feasibility, cost and time of the project. This may not be always correct as many costs are revealed during development, such as advancement in technologies may lead to redesign cost.

The problem with cost estimation is due to inaccuracy of models, incomplete requirements, lack of information on past projects and methods developed for specific data do not easily get adaptable to other environments [1][2][4].

II. SOFTWARE COST ESTIMATION & ITS IMPORTANCE:

Software Cost Estimation is the collection of methods and techniques to derive the software cost estimate. Software cost estimation came in 1940s with the beginning of computer industry and has been developed several times till the formulation of function points by Albrecht in 1979.

Developing the software for accurate cost estimating is crucial. Management may get convinced to develop new systems that may later result in budget overrun due to underestimating costs and may fail to meet their expected goals. Reason for cost overrun is poor estimates which further leads to wastage of resources invested in them. In fact the projects which begin to overrun are cancelled before their completion. [6]

After 40 years of research soft estimation remains to be a weak link in project management, it is still one of the challenging problems in software engineering. Predicting the size of the software system and its development effort is one of the biggest problems faced by the software developers and users. Unfortunately there is an inaccuracy when estimating cost and schedule in software industry [1][2].

Project manager is responsible for making accurate estimates early in the project, early estimates are difficult to obtain at an early stage of project development due to lack of

detailed information available. However, early estimates are required for determining feasibility of a project in terms of a cost-benefit analysis.

Several cost estimation tools or methods are difficult to interpret to be of much help in estimation process. Estimators rely on their past experience for predicting the effort for software projects. So, developing cost estimation methods which are easy to interpret, calibrate and use is of great need [16].

Significant research on software cost modeling began in 1965, contributing to some useful partial models in the late 1960s and early 1970s. In 1960, the concept of parametric estimation models was developed by Frank Freiman which lead to the development of PRICE model, Barry W. Boehm's Constructive Cost Model(COCOMO)[7], and Larry Putnam's Software Life Cycle Management (SLIM)[8] also came in existence in late 1970's performing estimates at various points early in a project is often error-prone, tedious, or just impossible to perform [16].

A. IMPORTANCE[9]

- It helps to distinguish and prioritize projects with respect to business plan. It leads to overall improvement.
- It helps in managing and controlling of software development projects by matching the resources with the real needs or requirements.
- As far as customer expectations are concerned, actual development cost should be almost same as the estimated cost.
- It helps to determine the required resources for the project and their utilization.
- It defines the resources to produce and verify software products and managing the activities during the software development.

III. ISSUES RELATED TO SOFTWARE ESTIMATION

Since the organization works solely on the basis of its experience when extending a commitment for a certain project to a client, it faces many hindrances during the project development life cycle. For example, after taking on a particular software project, the organization determines its complexity and size and considers its features to be implemented i.e., qualitatively.

This results in an incorrect estimation of total effort involved in the application development. The incorrect effort-estimation consequently results in the incorrect allocation of

human resources in the project, which further leads to non-compliance of project scheduling as well as cost overruns.

Other issues in consideration are as follows [12]:

- Somerville says that an early accurate effort estimate is to be based on high level user needs in order to develop a software system, as people involved and their skills are not known initially [10].
- McConnell states that software development is the process of refinement which results in the image of fuzzy software being developed as a result of which effort and time estimates needed to calculated [11].
- The approach of expert judgment is problematic [13]:
 - Highly experienced estimators are not available easily for every new project.
 - When size of the project increases cost increases with it, expert judgment method to be of use is only when the size of current running project is similar to the past comparable project.
 - This method does not give accurate results.
- Lack of available historical data for estimation.
- If the estimation is not apt little penalty is associated with it.

Simple methods available are not safe to use[14], these are based on cost and productivity rates of past projects, if new project does not have any cost critical differences from previous projects then this approach is suitable. But if some critical cost drivers are ignored then it is not safe. There is need to think about deviation b/w predicted formula and observed data.

IV. REASONS OF INACCURATE ESTIMATIONS

- Performance analysis rarely considers whether estimates are met or not.
- Lack of appropriate knowledge experts available.
- Improper methodology and guidelines for estimation.
- Lack of team coordination between various phases and functions during development.
- Lack of historical data available for comparison.
- Insufficient time for testing and analysis while developing estimates.

V. PROBLEM RELATED TO PREVIOUS METHODS

In order to use Non Algorithmic methods for cost estimation, historical data is essentially required on basis of which experts are able to judge the cost and size estimates for the current project. But as the projects in industry are becoming advanced and complex, organizations cannot depend for estimates upon this method of expert judgment as it does not give an accurate result.

Moreover when estimates are not up to the mark, organization faces the problem of cost rising and several other penalties. Even, when the project gets started initial estimates are required on the basis of which feasibility of the project is decided, incorrect estimates may even lead to failure of project before starting.

Algorithmic methods which include COCOMO, SLOC, Function Points, etc are based on mathematical calculation and experimental results [17] which are hard to learn and require lots of data, as the size of the project increases complexity of the project also increases which makes the estimation more problematic.

Another major problem which can lead to project failure in early stages is not identifying risk factors early during software development, this kind of security is not provided in any of the previous models, and there is a need to extend the current software cost estimation technique that may help the software developers to estimate the cost of the project more precisely [15].

For Example: The small industries having low budget and manpower, cannot afford to have their estimates using high cost and complex tools.

VI. PROPOSED SOLUTION

In order to ease these problems there is a need to build a hybrid tool which must have following characteristics:

- Enables the organizations to attain a wide range of input in order to give appropriate results to their clients.
- Enables the project manager to estimate the project size in terms of SLOC through function points and operational constraints.
- Helps the companies to bifurcate the total duration amongst SDLC phases, which results in a timely tracking of the overall progress of the project.
- Guides organization about the proportion of total estimated effort involved within different phases of the development life cycle of the project, which helps them in adopting necessary measures to plug any possibility of schedule or cost overruns.

- Tool must provide an additional database for storing the results and data related to previous estimated projects which can help the organization for future references.

All in one this tool can assist commercial software development organizations to quantitatively estimate analyze and utilize fundamental software metrics that are a key to the success of any project.

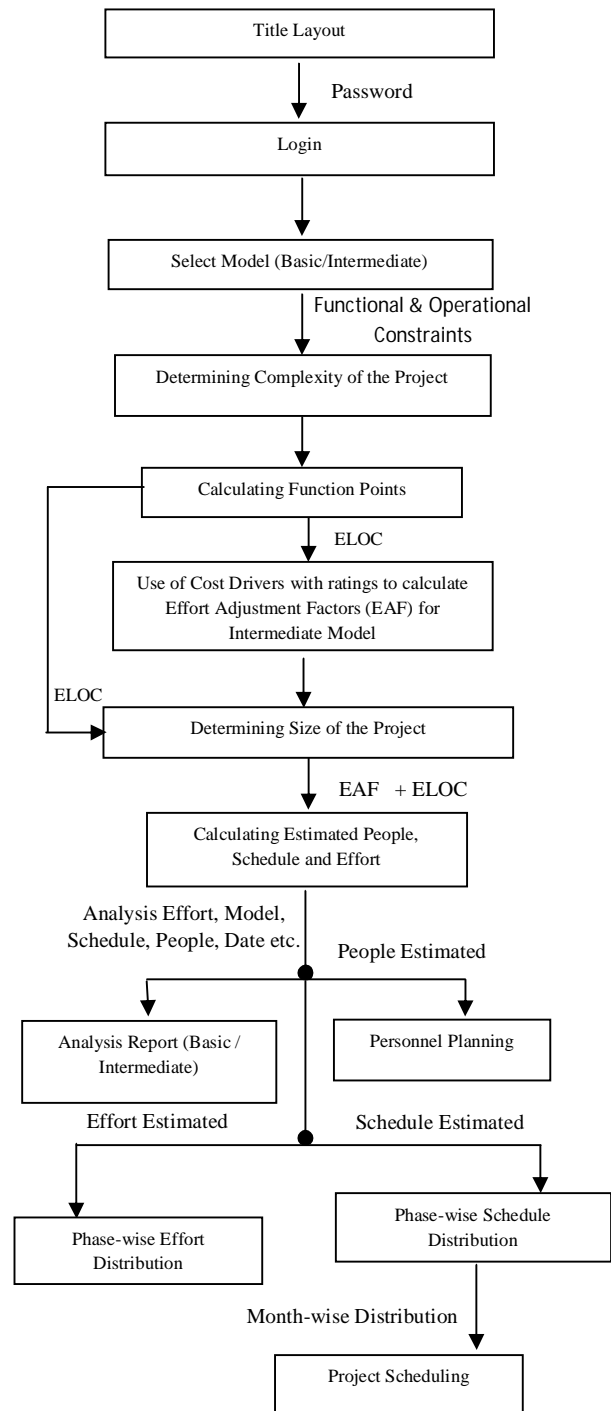


Figure 1: Step-wise working of proposed cost estimation tool

VII. WORKING OF TOOL

A. Determining the complexity

The determination of the complexity is significant in order to evaluate the total effort and time for its implementation. Factors need to be considered for complexity determination is:

- External input must be monitored continuously for internal evidence, consistency and reliability.
- Complexity and data structures used in the project are the most important factor.

We will classify the project complexity into five broad categories: Very Simple, Simple, Average, Complex, and Very Complex.

Once the appropriate complexity is selected by the Project Manager, a numeric variable is initialized according to his/her selection of project's class or complexity, i.e. the higher the complexity, the higher the value of the initialized variable, and vice-versa.

B. Determining the Size

In order to calculate the size of the project, our application would enable the user to examine it from three different dimensions: Functional characteristics, Operational Constraints, Programming Language.

C. Determining Cost Factors and Estimates of Effort, Time and People

The four important attributes that enable Project Manager to determine cost drivers are mentioned below:

- **Product Attributes:** These factors signify the important features of the product, such as reliability, complexity, etc.
- **Hardware Attributes:** These reflect how advanced hardware configuration are required in order to implement the solution.
- **Personnel Attributes:** These attributes would help in rating the features related to human resources like efficiency, productivity, competency, programming capability, experience, etc.
- **Project Attributes:** These signify project's characteristics such as milestones, deliverables, etc.

User rates the attribute while taking complexity and size into consideration, higher the rating, higher is the amount of estimation of effort, people and time required to develop the project.

To calculate the total estimated Effort, Time, People and Cost, we will use size, complexity and inputs of the solution. We plan to calculate the size by considering the functional and

non functional requirements, operational constraints and complexity.

Further, we have implemented two models of COCOMO which are *Basic & Intermediate*, respectively.

D. Determining Project Planning and Management

After determination of estimated Effort, People, Cost and Time required for project implementation, we can utilize these metrics for planning and managing projects. So, we can plan the implementation by knowing the proportion in which metrics should be applied in various phases of Software Development Life Cycle.

If we know what portion of the estimated effort will be applicable in various phases of the software life cycle (such as the Analysis, Architectural Design, Detailed Design, Coding, Testing and Deployment phases of the desired implementation), we can plan the entire project and its corresponding milestones accordingly. We can also plan the number of people that should be designated for the timely completion of a particular phase.

In our application, we have divided the total estimated effort amongst various development life cycle phases of the project according to the proportion that is dependent upon the solution's complexity and pre-defined research percentages

E. Determining Effort and Schedule Distribution

The distribution of estimated time depends upon the complexity and size of the application in addition to the proportionate effort in various phases of the development life cycle. The estimated effort is related to the complexity and size of the proposed application, and its distribution is also related in the same way. The distribution of total time depends on the effort-distribution of the application in various phases; this means, the greater the effort, the more is required. Hence, once we have the distributed timelines, we can effectively plan the milestones of the applications.

This division eventually helps us to plan, schedule and track the whole project in terms of milestones while identifying any scope of possible improvements that may exist in our estimation procedure or application of effort under software development life cycle phases in order to implement the project or module efficiently and without delays.

F. Project Scheduling

Project planning is extremely vital to the successful, on-schedule completion of the project because it reflects process-maturity in an organization and provides the opportunity for further improvement. One of the subset phases of planning is tracking, which involves a verification of the total implementation progress as per the defined timelines or milestones. If there is a lag, it gives us an opportunity to identify any possible improvements in our software

engineering practices or software estimation and measurements procedures.

Furthermore, if the team discovers that they are deviating from the defined schedule, it will give them the opportunity to identify any existing possible loopholes and corresponding process-improvements or changes in the estimation procedure through considering any additional, hidden functional or operational characteristics.

G. Database

Now our next objective is keeping record of estimated data which would be useful for further analysis and comparison of future projects. For example: if the organization wins a project which has the same requirements as one of the previous projects done by the organization then if the estimated data from the previous implementation is stored, it will take less time for organization to plan and schedule the implementation as compared to previous one.

We have done the implementation of recorded data in the form of report containing every information regarding analysis of previous projects. Details like project class, estimated effort, estimated time, estimated persons and analysis timestamp are included in the report; it creates detailed, convenient and near-precise analysis inferences.

VIII. CONCLUSION

In this paper, we have proposed and developed an estimation tool for software industries in order to improve the estimation process. It is intended for organizations that need to determine various measures such as complexity, size, effort, manpower, cost, etc which are involved in project during planning phases. The distinguishing features that separate existing software estimation tools from this application are its hybrid implementation methodology of estimating software metrics (such as Effort, Time and People) and its inbuilt capabilities of planning human resources, storing data for further analysis and scheduling a project. Furthermore the estimates done by the tool must be accurate and informative in terms of project planning, management and should provide convenience to the organization in fulfilling the requirements of their clients. Tool should empower the organization's maturity and goodwill and thus should make out accurate results.

REFERENCES

- [1] Linda M. Laird, "The Limitations of Estimation," IT Professional, vol. 8, no. 6, pp. 40-45, Nov./Dec. 2006, doi:10.1109/MITP.2006.149
- [2] Barry Boehm, "Safe and Simple Software Cost Analysis," IEEE Software, vol. 17, no. 5, pp. 14-17, September/October, 2000

- [3] A. Issa, Algorithmic software cost estimation model for early stage of software development, international journal of academic research, March 2011
- [4] Software Development Cost Estimation. Phillippe Kruchten, University of British Columbia
- [5] K. Kavoussanakis and T. Sloan. UKHEC Report on Software Estimation, Dec 2001, accessed on <http://www.ukhec.ac.uk/publications/reports/estimation.pdf>
- [6] A.L. Lederer, and J. Prasad, Causes of Inaccurate Software Development Cost Estimates. Journal of Systems and Software, pp. 125-134, 1995
- [7] B. Boehm. Software Engineering Economics. Prentice Hall, 1981.
- [8] L. Putnam and W Myers. Measures for Excellence. Yourdon Press Computing Series, 1992.
- [9] Zia, Z.; Rashid, A.; uz Zaman, K.; "Software cost estimation for component based fourth-generation-language software applications," Software, IET , vol.5, no.1, pp.103-110, February 2011 doi: 10.1049/iet-sen.2010.0027
- [10] I. Sommerville. Software Engineering, Sixth Edition. Addison-Wesley Publishers Limited, 2001.
- [11] Steve McConnell. Rapid development: taming wild software schedules. Microsoft Press, 1996.
- [12] H. Leung and Z. Fan, Software Cost Estimation. Handbook of software engineering and knowledge engineering, world scientific publications company, River Edge, NJ, 2002.
- [13] M. Jorgensen and S. Grimstad: "Software Development Effort Estimation: Demystifying and Improving Expert Estimation.
- [14] Barry Boehm, "Safe and Simple Software Cost Analysis," IEEE Software, vol. 17, no. 5, pp. 14-17, Sep./Oct. 2000, doi:10.1109/52.877854
- [15] N. Abdullah and R. Abdullah: "Potential Security Factors in Software Cost Estimation," International Symposium, Vol. 3, page 1-9, 26-28 Aug. 2008.
- [16] A. Zaid, Mohd H. Selamat, Abdual Azim Abd Ghani, R. Atan and Tieng Wei Koh: "Issues in Software Cost Estimation".
- [17] V.Khatibi and D.Jawawi, Software Cost Estimation Method: A Review. *Journal of emerging trends in computing and information sciences*, 2011