ANALYSIS OF SOFTWARE COST ESTIMATION MODELS

Tahir Abdullah  
GC University  
ahir_wains@hotmail.com

Rabia Saleem  
GC University  
rabia_278@yahoo.com

Shahbaz Nazeer  
GC University  
shahbaz_nazir1@yahoo.com

Muhammad Usman  
GC University  
email2usman@gmail.com

Abstract - Software Cost estimation is a process of forecasting the Cost of project in terms of budget, time, and other resources needed to complete a software system and it is a core issue in the software project management to estimate the cost of a project before initiating the Software Project. Different models have been developed to estimate the cost of software projects for the last several years. Most of these models rely on the Analysts’ experience, size of the software project and some other software metrics. Each model has its some key strengths and some limitations. In this research some effort estimation models are assessed for their strengths and limitations. This research will propose some effective measures to select appropriate model in appropriate mode of software development.

Key Words: Software Cost Estimation, Cost Estimation Models, Effort Estimation, Software Project Estimation.

I. INTRODUCTION

Software Cost estimation is a process of predicting and measuring the efforts and cost required developing a software project before the start of the software project. Now a day Software Cost Estimation has become a complicated task in software project management field. Accurate Cost estimation is very critical issue for the software Developing organizations. Number of models has been developed to estimate the effort and cost of software projects for the last several years. Most of these models rely on the size of the software projects and different software metrics. Each model has some key strengths and some limitations. In this research some cost estimation models are assessed for their strength points and limitations.

“The software cost estimation is an important task within projects. It determines the success or failure of a project. In order to improve the estimation, it is very important to identify and study the most relevant factors and variables”. (Mannequin et. al. 2005) [1]

Soft Cost Estimation involves the determination of one or more of three things i.e. Effort (Usually in Person-Months), Project Duration (In Calendar Time) and finally Cost (In Dollars). Most cost estimation models attempt to generate an effort estimate, which can then be converted into the project duration and cost (Leung and Fan 2002) [2].

The core part of the software project development is to appropriately estimate the cost for a project which further provides the easy way to the other project activities. The core parts should be included in the cost estimation list which are given below

1. Project domain
2. Technical resources for development.
3. Financial issues
4. Well scrutinized method for distribution of finance

Zaid et. al. (2008) [3] described metrics involved in software cost estimation. These metrics are size of software, complexity and some other metrics and other issues regarding these metrics in different models.

Many software development organizations track and analyze project concert by measure cost estimation correctness. A high Cost Estimation error is regularly interpreting as poor estimation skills. This is not of necessity a correct understanding. High cost estimation error can also be a result of other factors, such as high opinion difficulty and deficient cost control of the project. Through a real-life example we exemplify how the lack of good estimation error analysis technique can bias analyses of cost estimation correctness and lead to wrong conclusion. Further, we look at a selection of cost estimation studies, and show that they regularly do not take the needed actions to ensure significant interpretation of estimation error data. Aggravated by these results, we propose a general framework that, we believe, will improve analyses of software cost estimation error (Grimstad and Jorgensen, 2006) [4].

II. COST ESTIMATION MODELS

There are several models of Software Cost Estimation, some of which uses Source Lines of Code as input and base on SLOC e.g. COCOMO. On the other hand some of the models are based upon the functionality of the software to be developed e.g. Function Point. Some of the models depend upon the experts’ judgments e.g. Delphi Technique. No model is best in some certain condition, situations and environments (Basha and Dhavachelvan 2010) [5].

According to (Leung and Fan 2002) [2] There are basically two types of Models: Non-Algorithmic
Models, Algorithmic Models vary widely in sophistication. Non-Algorithmic Models include SLOC, Expert Judgment, Parkinson, Price-to-Win etc. Algorithmic Models include COCOMO, Function Point (FP) etc.

2.1 Source Line of Code (SLOC)

Source Line of Code Technique (SLOC) is an old and considered unreliable technique to estimate the cost of the project due to various reasons. The major drawbacks of this model are that it depends mainly on the experience of the analysts, and language on which software is to be developed. The Estimated Project Size helps determine the resources, effort, and duration of the project. There are many techniques to calculate the size of a software project. You can calculate the size by using a directly measurable technique, the SLOC technique. It is defined as the source lines of code that are delivered as part of the product. The effort spent on creating the source lines of code is expressed in relation to thousand lines of code (KLOC). The SLOC technique is an objective method of estimating the size because there are no multiple ways of calculating the lines of code. Therefore, the effort estimate is close to being accurate.

This technique includes the calculation of lines of code, documentation of pages, inputs, outputs. And components of a software program. Components are again of multiple types, such as reports. Screens and files. The SLOC technique is not used so much to directly calculate the effort to be spent on a project, rather it supports some other models to estimate the cost of Software Project. A typical method for calculating the estimated SLOC is PERT (Leung and Fan 2002)[2]. It involves Expert judgments’ of possible three sizes Sl, the lowest possible size; Sh the highest possible size; and Sm, the most likely size. The estimate of the code-size S is computed as:

\[ S = \frac{(SL+SH+4SM)}{6} \]

2.2 Function Point Technique (FP)

Function Point Model is an Algorithmic Model which is based upon the functionality of the software to be developed. The strengthening point of this model is that it covers the problems of nature of language in which software is to be developed. It does not rely highly on the experts’ judgment. But at the same time it is a drawback of this model that consideration of the estimated size and expert’s judgment is an important part of cost estimation which this model doesn’t support.

Using Function Point technique first information domain of the software is weighted according to the given table

<table>
<thead>
<tr>
<th>Information Domain</th>
<th>Low</th>
<th>Medium</th>
<th>High</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Inputs</td>
<td>x3=</td>
<td>x4=</td>
<td>x5=</td>
<td>x6=</td>
</tr>
<tr>
<td>User Outputs</td>
<td>x4=</td>
<td>x5=</td>
<td>x6=</td>
<td>x7=</td>
</tr>
<tr>
<td>User Inquiries</td>
<td>x3=</td>
<td>x4=</td>
<td>x5=</td>
<td>x6=</td>
</tr>
<tr>
<td>Internal Files</td>
<td>x7=</td>
<td>x10=</td>
<td>x15=</td>
<td>x15=</td>
</tr>
<tr>
<td>External Interfaces</td>
<td>x5=</td>
<td>x7=</td>
<td>x10=</td>
<td>x10=</td>
</tr>
</tbody>
</table>

Table 1: Information Domain Weight Tables

Each type of software has some specific characteristics. These characteristics are defined in a table given.

<table>
<thead>
<tr>
<th>Sr. #</th>
<th>Characteristics</th>
<th>Degree of Influence Value (0 to 5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Data Communications</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>Distributed Functions</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>Performance</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>Heavily used Configuration</td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>Transaction Rate</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>Online Data Entry</td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>End-user Efficiency</td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>Online update</td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td>Complex Processing</td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td>Reusability</td>
<td></td>
</tr>
<tr>
<td>11.</td>
<td>Installation ease</td>
<td></td>
</tr>
<tr>
<td>12.</td>
<td>Operational Ease</td>
<td></td>
</tr>
<tr>
<td>13.</td>
<td>Multiple site</td>
<td></td>
</tr>
<tr>
<td>14.</td>
<td>Facilitates Change</td>
<td></td>
</tr>
</tbody>
</table>

Total GSCs. \( \sum F_i \)

Table 2: Calculation of Total GSCs

After completing these table value adjustment factor is calculated as

\[ \text{Value Added Factor (VAF)} = (0.65 + [0.01 \times \text{sum of } F_i]) \]

Function Points (FP) = TUF \times VAF

Hence Function Points are obtained in Person Months. And after that Total Cost is calculated as defined Person Months Cost.

2.3 Constructive Cost Model (COCOMO)

COCOMO is a cost estimation which was propounded by Dr. Boehm (1981)[6]. This technique which considers functionality, different attributes of the system as well as size of the project also. It also
considers experts’ judgment which was deficiency of FP method. There are basically three levels of COCOMO i.e. Basic, intermediate and Advanced. Asoke (2004) [7] described the three level of COCOMO in details.

A) Basic COCOMO

The basic COCOMO technique estimates the efforts and cost of software project by using only Lines of Code. According to Basic COCOMO there are three types of projects to calculate effort: Organic, which have sufficient and defined objectives and mostly are simple business and financial applications. e.g. Banking and Inventory Systems etc. Embedded, this uses hardware as well as software. Such projects include Real-time Operating Systems, Industrial Automation etc.

Estimating the effort in Basic COCOMO involves three steps.
2. Determining the efforts constants based on the type of the project as given in Table 3.

<table>
<thead>
<tr>
<th>Project Type</th>
<th>a1</th>
<th>a2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organic Projects</td>
<td>3.2</td>
<td>1.05</td>
</tr>
<tr>
<td>Embedded Projects</td>
<td>2.8</td>
<td>1.20</td>
</tr>
<tr>
<td>Semidetached Projects</td>
<td>3.0</td>
<td>1.12</td>
</tr>
</tbody>
</table>

Table 3 (Effort Constraint Table)

3. Substituting values for lines of code and effort constants in formula to calculate effort in Person Months.

\[
E_i = a_1 \times (KLOC)^{a_2} \quad \text{(In Person Months)}
\]

B) Intermediate COCOMO

Intermediate COCOMO involves some of the other factors called Cost Driver Attributes, which are not covered in Basic COCOMO. And through these Cost Driver Attributes Effort Adjustment Factor (EAF) is calculated. These Cost Driver Attributes relate to different aspects of Software. Steps involved in Intermediate COCOMO are

1. Estimate the effort \(E_i\) (through Basic COCOMO).
2. The second step is to find out the applicable cost driver attributes that involve the project intensively. This provides value for EAF for project.

\[
EAF = \prod E_i \quad \text{(Product of intensities of all applicable cost driver attributes)}
\]

\[
E = EAF \times E_i \quad \text{(In Person Months)}
\]

C) Advanced COCOMO

The advanced COCOMO technique uses the steps of the intermediate COCOMO technique. In addition, it uses costs driver attributes assigned to each phase of the SDLC such as analysis and design.

This research basically focuses on the comparison of algorithmic Models Function Point and COCOMO for Software Cost Estimation. Different nature of projects of different sizes was considered to be analyzed. These projects were developed using different Technologies under supervision of developers of different experience level.

III. VARIABLES

Prediction of the development effort has always been a big challenge for a software system based on developer abilities, size, complexity and other metrics for the last decades. The ability to give a good estimation on software development efforts is required by the project managers (Attarzadeh and S. Hock, 2009) [8]. According to Smith (1997) [9] there are several metrics which can affect the cost estimation. Some of them are Fragmentation, Component Programmer Experience, Programmer project experience etc.

There are various parameters discuss in this study.
1. Major Technology/Language Used.
2. Size of Project (In KLOC).
3. Software Analyst Experience (In years).

Copyright © 2012 IJEIR, All right reserved
4. Software Developer/Programmer Experience (In years).
5. Application Experience.

1) Major technologies
Cost of software project depends on the major tools and technology used as different technology very in complexity and there features. And also variation in availability of the expert persons of these technologies. The technologies considered for this study were
   1. Oracle + Developer
   2. Visual Basic with some DBMS.
   3. Dot net with Some DBMS.
   4. PHP with MySQL.

These technologies were selected for each type of software development.
   1. Distributed Development Environment (Oracle Developer).
   2. Desktop Application (Visual Basic).
   3. Online Systems/ Website Development (PHP, MySQL).
   4. Online and Desktop Applications (Dot net).

Software Cost Estimation depends upon different software Metrics. The metrics considered in this study were Size of Project, Analyst’s Experience, Developer’s Experience and Relevant Application Development Experience. And these metrics were assessed and grouped as given.

2) Size of Project (In KLOC)
   1. Small (5-8) KLOC
   2. Medium (9-10) KLOC
   3. Large (10-15) KLOC
   4. Extra Large Above 15 KLOC.
   5. Projects Less than 5 KLOC were not considered.

3) Analyst Experience
   1. Low Having Experience Less Than 3 year
   2. Medium Having Experience 3-6 year
   3. High Having Experience more Than 6 year.

4) Developer Experience
   1. Low Having Experience Less Than 3 year
   2. Medium Having Experience 3-6 year
   3. High Having Experience more Than 6 year.

5) Relevant Application Experience
   1. Low Having Experience Less Than 2 to 3 Application
   2. Medium Having Experience 5 to 8 Application
   3. High Having Experience 10 to 15 Application

IV. PROCEDURE

A survey was conducted to elicit the different parameters of both the techniques of Function Point and Constructive Cost Model. Through which the estimated costs of the Software Projects under consideration were assessed by applying both the techniques. The actual costs of these projects were compared to the estimated costs calculated by these methods to draw the results.

IV.1 Stack Holders

The stakeholders who were considered for the survey for the elicitation of required parameters were
   • Project Manager
   • Senior System Analyst

This survey was conducted to elicit information from project manager, Senior System analyst of software industries and software considered for the survey were of different nature, like In-house development as well as software houses also considered. The results were obtained and graphs were drawn after cross tabulation of variables.

V. RESULTS AND DISCUSSIONS

A) Comparison of Techniques according to Major Technologies used:

Technologies/L0languages can affect the cost estimation of a software project in both the techniques. But the effect can be different in percent deviation from actual cost. For COCOMO technique the ratio of high percent deviations is decreased for almost all technologies/languages used as shown in fig 1(a). In case of FP method ratio of percent deviation from actual cost is uncertain and it increases for all technologies used as shown in Fig 1(b).

![Affect of Technology on Deviation from Actual Cost using COCOMO](image-url)
B) Comparison according to size of the project
Size of the project also affects in both techniques in a different way. COCOMO is not much appropriate for small level projects as ratio of high deviation from actual cost increases up to a certain level. But it is appropriate for medium, large and extra large projects as shown in Fig 2(a). On the other side FP is unreliable in case of size also. Because its ratio of deviation also increases for almost all sizes. But it is to some extent better for large and extra large in comparison with small as shown in Fig 2(b).

C) Comparison according to Analyst Experience
In COCOMO as the experience of the Analyst increases the ratio of percent deviation from actual cost decreases and in medium level of experience it is uncertain as shown in Fig 3(a). In case of FP method percent deviation from actual cost increases whether experience is high or medium. In Case of high experience its increase ratio is less than medium as shown in Fig 3(b).
D) **Comparison according to Developers Experience**

Software Developers’ capabilities and experience also affect the cost of the project. Cost estimation can also deviate from the actual cost due to Developer’s experience level. In COCOMO as the experience of the developer is increased the deviation of cost from actual decreases as shown in Fig 4(a). In FP method deviation increases either experience is medium or high with another fact that in case of high experience the rate of decrease is relatively low as shown in Fig 4(b).

E) **Comparison According to Relevant Application Experience**

In COCOMO if the relevant application experience is low then the percent deviation from actual cost increases but as the level of experience increases to medium and then high, the percent deviation decreases and overall with increases in relevant application experience decreases the percent deviation from the actual cost as shown in Fig 5(a). In case of FP Only with High level of relevant experience the percent deviation from actual cost can be decreased but that is also very minor as shown in Fig 5(b).
IV. CONCLUSION

Software Cost Estimation is one of the crucial matters in software development. This research focuses the behavior of Software Cost Estimation Models according to different factors. Basically two different types of models were compared and analyzed in this research. It is fact that we can’t measure the exact cost of the software to be developed; there is always some deviation of estimated cost from actual cost. But using appropriate cost estimation model this deviation can be minimized. This deviation differs in impact in different Technologies, size of project and some other described factors while using these models. But after comparison it was analyzed that COCOMO model is overall better technique because of its features of considering SLOC and Cost Driver Attributes.

REFERENCES


AUTHOR’S PROFILE

Tahir Abdullah

He is presently serving as Lecturer in College of Computer Science & Information Studies, Government College University Faisalabad, Pakistan. He is MS (CS) with specialization in Software Engineering from University of Agriculture, Faisalabad and M. Sc (CS) with Specialization in Software Engineering from University of The Punjab, Lahore. His research areas are Software Engineering, Aspect Oriented Software Engineering and Risk Management.

Rabia Saleem

She is presently serving as Assistant Professor in College of Computer Science & Information Studies, Government College University Faisalabad, Pakistan. She is MS (CS) from University of Central Punjab Lahore, Pakistan and BS (CS) from Muhammad Ali Jinnah University, Karachi, Pakistan. Artificial Intelligence and Human Computer Interaction are her areas of specialization.

Shahbaz Naeer

He is presently serving as Lecturer in College of Computer Science & Information Studies, Government College University Faisalabad, Pakistan. He is MS (CS) with specialization in Software Engineering from GC University, Faisalabad Pakistan and BS (CS) (Gold Medalist) with specialization in Software Engineering from GC University, Faisalabad Pakistan. His research areas are Software Engineering, Artificial Intelligence and Software Requirement Engineering.

Muhammad Usman

He is presently serving as Lecturer in College of Computer Science & Information Studies, Government College University Faisalabad, Pakistan. He is MSc (CS) from University of The Punjab Lahore, Pakistan. His research areas are Software Engineering and Software Design and Architecture.