MIEL Experience in Calibration of COCOMO II Post Architecture Model

Santha Kumar V A
Motorola India Electronics Ltd.
"The Senate", # 33A, Ulsoor Road,
Bangalore - 560 042, INDIA.
Email: shantha@miel.mot.com
Phone: 91-80-5598615
Fax: 91-80-5598660

Abstract

This paper describes experience of Motorola India Electronics Limited (MIEL), Bangalore in Calibration of COCOMO II Post Architecture Model. Effort estimation in MIEL is being done using the Wide Band Delphi (WBD) method. Typical estimation accuracies are of the order of plus or minus 20%. There was a strong business need to use a cost estimation model to validate the estimates obtained using WBD method. COCOMO II was considered for pilot study as it incorporates the use of various drivers impacting software development in its model (e.g. domain expertise, Complexity, Process Maturity, Reuse, Personnel Continuity etc.) and it also allows the users to calibrate the database to suit their needs. The results from the direct application of COCOMO II was still not satisfactory, our goal was to improve estimation accuracy to plus or minus 5%, hence calibration of COCOMO II was taken up as a next step to obtain better estimation accuracy. In this paper, the approach taken to calibrate the model using multiple Regression Analysis, the shortcomings of using this model and the results of the calibration are discussed.

1 Introduction

Software Cost Estimation continues to be one of the challenging areas in software project management. For better project management, projects need to accurately predict effort required. This is a very difficult proposition as the problems faced in effort estimation are dependent on size estimates, complexity of projects, and requirement volatility. Following approaches are available for predicting effort

- Expert judgement
- Algorithmic Models
- Machine Learning

In the area of cost estimation, majority of research has been carried out on algorithmic models whereas expert judgement is the widely used technique. In this technique a combination of individual's domain expertise, data on similar kind of development and baseline data of completed projects are used to come out with estimates. Wide Band Delphi (WBD) is a structured approach of expert judgement. With the traditional statistical approach based models not being able to cope with problems in estimation like missing value, outliers in data, there have been research in non-traditional areas like Fuzzy Logic, Neural Networks, Analogy, Regression tree etc., these fall under the machine learning technique
umbrella. In the area of Algorithmic Models researches have been carried out widely. The best known among the algorithmic models are COCOMO [boehm81], SLIM and Function points [Albrecht83]. At MIEL WBD method is widely used. Historical data on the estimation is maintained across projects from various domains. Accuracy data is available on schedule estimation, size estimation and effort estimation. In the year 1997, effort estimates were within 5% of actual 22% of the time and within 20% of actual 59% of the time.

With effort estimates playing a very important role in projects based on fixed cost, there is a need to improve the EEA figure of 22% for projects estimating effort within 5% actual. This was a motivation for looking at a cost estimation model for providing an alternate estimate apart from being useful in validating the WBD estimates. COCOMO was chosen for use, as it is transparent unlike other proprietary models such as SLIM. This gives the user flexibility to calibrate the model using historical data to achieve better results. Also COCOMO incorporates the use of various drivers impacting software development in its model (e.g. domain expertise, Complexity, Process Maturity, Reuse, Personnel Continuity etc.).

1.1 Overview of the model:
The COCOMO II Post Architecture model has the following form

\[
\text{Effort} = A \times \{\text{size}\}^B \times \prod_{i=1}^{17} \text{EM}_i
\]

Eq. 1

Where,

- \( A \) = Multiplicative Constant
- \( \text{Size} \) = size of software in KSLOC
- \( B = 1.01 + 0.01(SF_1 + SF_2 + \ldots + SF_5) \), where \( SF \) = Scale Factor
- \( \text{EM} \) = Effort Multiplier (refer [Manual97] for details)

The 17 effort multipliers or cost drivers are stratified into four groups' viz., Product, Platform, Personnel and Project. The cost drivers have ratings from very low to extra high, for further details on the range for individual cost drivers refer [Manual97].

1.2 Scope of Calibration
Calibration was done only for the exponential constant “B”.
Calibration was done and results analyzed only for Total Effort in Staff-Months.
Overall Schedule and Phase wise Effort Estimates not studied.

2 Data Collection & grouping mechanism
For the purpose of this study, the project groups were stratified into different sub groups based on the domain. All projects developing Test Tools were clubbed into one group (called Test) and projects having development based on OO methodology formed another set (called OO).
All the projects falling into these groups, which were completed in 1996 and 1997, filled out the COCOMO II questionnaire. The data included, apart from the COCOMO II cost driver ratings, actual effort that was expended in the project and the actual size.

3 Initial Results and Approaches taken for calibration

3.1 Interpretation of Cost Drivers:
Personnel Cost Drivers like AEXP, PEXP, LTEX had interpretation different from that of COCOMO definitions. Experience shows that a person working in a particular domain continuously for a couple of years becomes an expert in that domain. So the ratings were accordingly interpreted. For e.g. PEXP had the following rating

<table>
<thead>
<tr>
<th>COCOMO definition</th>
<th>Very Low</th>
<th>Low</th>
<th>Nominal</th>
<th>High</th>
<th>Very High</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>≤ 2 months</td>
<td>6 months</td>
<td>1 year</td>
<td>3 years</td>
<td>6 years</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MIEL interpretation</th>
<th>&lt; 2 months</th>
<th>2 month to &lt; 6 months</th>
<th>6 months</th>
<th>6 month to &lt; 2 years</th>
<th>≥ 2 years</th>
</tr>
</thead>
</table>

These interpretations of cost drivers were done consistently across while doing the rating for different projects.

3.2 Initial Results
Data collected using the COCOMO II questionnaire was directly applied on the COCOMO II equation. The results obtained showed that the estimates were within 40% of the actual only 20% of the time for 'Test' and it was 36% for 'OO'. This result prompted us to look at calibration of COCOMO II because, as per organizational baseline figures, generated using historical data across all domains, effort estimates were within 20% of the actual 59% of time in the year 1997.

3.3 Least Square Method
The approach taken in calibrating COCOMO II was to apply the Least Square Method (LSM) as described in many statistical textbooks like [gupta97] and the calibration approach using LSM as described in [boehm81], as a first step of applying Multiple Linear Regression (MLR). LSM was applied on each data set, Test & OO, separately to find out values of 'B'. The value of 'B' got using LSM was used in MLR method.
3.4 Multiple Linear Regression

In Multiple Regression, several regressors are used to model a single response variable [Weisberg85]. It is based on Least Square method. The model is expressed as

\[ Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \xi \]  

Eq. 2

Where, the \( \beta \)'s are the unknown parameters, \( Y \) is the response, \( \xi \)'s are statistical error and is normally distributed and \( X_1, X_2, \ldots, X_p \) are the regressors. In MRL the least square estimate \( \beta' \) of \( \beta \) is chosen to minimize the residual sum of squares (RSS). Suppose, if \( X_i \) be the \( i^{th} \) row of \( X \), \( i=1, \ldots, n \). We choose \( \beta' \) to minimize the function

\[ \text{RSS}(\beta) = \sum (Y_i - X_i^T \beta)^2 = (Y - X \beta)^T (Y - X \beta) \]  

Eq. 3

The least square estimate of \( \beta' \) of \( \beta \) is given by

\[ \beta' = (X^T X)^{-1} X^T Y \]  

Eq. 4

provided \( (X^T X)^{-1} \) exists.

The COCOMO II Post Architecture model has the following form

\[ \text{Effort} = A \cdot \{\text{size}\}^B \cdot \Pi_{i} \text{EM_i} \]  

Eq. 5

\[ A = \text{Multiplicative Constant} \]
\[ \text{Size} = \text{size of software in KSLOC} \]
\[ \text{B} = 1.01 + 0.01 \times (SF_1 + SF_2 + \ldots + SF_5), \text{where SF = Scale Factor} \]
\[ \text{EM} = \text{Effort Multiplier (refer [Manual97] for details)} \]

Since the restriction imposed by MLR on large number of data points relative to the number of model parameters, calibration of EMs was not attempted but the exponential factor 'B' was calibrated. So for a dataset containing 'm' projects we will have

\[ B_j = \beta_0 + \beta_1 S F_{j1} + \ldots + \beta_5 S F_{j5} \text{ where } j=1 \text{ to } m \]  

Eq. 6

As a starting point for applying MLR on the data set LSM was applied and estimates for 'A' and 'B' were obtained. Substituting this value for 'B' in Eq. 6 for all \( j \) we will get 'm' equations which can be represented in a matrix form, as

\[ B = [\beta_0, \beta_1, \ldots, \beta_5] \begin{bmatrix} 1 & 1 & \ldots & 1 \\ SF_{i1} & SF_{i2} & \ldots & SF_{im} \\ \vdots & \vdots & & \vdots \\ SF_{i5} & SF_{i6} & \ldots & SF_{im} \end{bmatrix} \]

where 'B' is a constant vector.
\[ B_{(1 \times m)} = \beta_{(1 \times 6)} \ast SF_{(6 \times m)} \]  

Eq. 7

MLR was applied on this Eq. and the co-efficient \( \beta \)'s were got. While estimating effort these \( \beta \)'s were used and also the effort estimation accuracy (EEA) were obtained.

4 Calibration Results

The results obtained from using the calibrated 'B' showed improvement over the initial results and it is as follows

**Table 4-1 : EEA for Data Set 'Test' with calibrated 'B'**

<table>
<thead>
<tr>
<th>Estimates</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Within 20% of actual</td>
<td>36%</td>
</tr>
<tr>
<td>Within 30% of actual</td>
<td>45%</td>
</tr>
<tr>
<td>Within 40% of actual</td>
<td>64%</td>
</tr>
</tbody>
</table>

**Table 4-2: EEA for Data Set 'OO' with calibrated 'B'**

<table>
<thead>
<tr>
<th>Estimates</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Within 20% of actual</td>
<td>27%</td>
</tr>
<tr>
<td>Within 30% of actual</td>
<td>36%</td>
</tr>
<tr>
<td>Within 40% of actual</td>
<td>45%</td>
</tr>
</tbody>
</table>
There is a marked improvement in the effort estimation accuracy when one uses the calibrated values for ‘B’ in Eq. 5. When more data points are added to the data sets the accuracy of the estimates are expected to improve.

5 Factors impacting results

The major factors impacting the results of calibration are lack of large number of sample and the complex nature of the calibration process. Lack of adequate number of projects meant that calibration of cost drivers could not be done. So calibration of the equation ‘B’ was taken up instead of the cost drivers. The calibration process by itself is complex as it involves advanced statistical techniques that are otherwise not normally used during the course of any software development.

6 Challenges in estimation

6.1 Size

Almost all the algorithmic models take size as input. When it comes to estimation accuracy, size estimation accuracy has more variation than the other two estimation accuracies. Lack of better control on software size estimation obviously impacts the effort estimation accuracy, more so when one uses these algorithmic models.

6.2 Language

Use of different languages in a project has an impact on the estimates, as one has to take into account the varying productivity levels for the different languages used. Use of new languages, like Java, also pose challenges as the project teams are not able to gauge the productivity of these languages. Since it is a new language in the market, industry figures are not immediately available.

6.3 Type of Development

With mounting competition organizations are looking at cycle time reduction in software development. To achieve cycle time goals organizations are looking beyond traditional method of hand coding. Organizations are going in for COTS, Reuse, Tool generation etc., so as to develop products with fast turn around time. Effort estimation becomes a difficult proposition when there is a project with these different types of development.
6.4 Maintenance projects

Despite the fact that maintenance costs are significantly higher than development costs, over life of the product, little attention is being paid for it. Study like the one conducted by [Vigder94] shows that very few organizations account for maintenance cost as part of the life cycle cost.

How does one overcome these challenges when using COCOMO II? If all these challenges cannot be overcome by an approach applying statistical technique for cost estimation, then the need of the hour is to look non-traditional approaches fuzzy logic or neural networks or chaos theory more seriously to provide solutions.

7 Conclusion

An effort has been made to use COCOMO II 1997 Post Architecture model to estimate effort. Since the results obtained by calibration of ‘B’ shows improvements over the initial results we can extrapolate and say that, if more data points are available and effort multipliers are calibrated to suit the needs of MIEL, then better results can be expected.

8 Acknowledgement

The author is grateful to Dr. R.Chandrasekaran, Professor, Department of Statistics, Madras Christian College, Chennai, for his guidance. The author also wishes to thank the members of the MIEL Software Quality Engineering department and the members of the projects, where this study was done, for their support.

9 Reference:


