RESEARCH ARTICLE OPEN ACCESS

An Evolution of Cost Productive Method for Software Reuse

Navdeep Singh[1], Sumeet Kaur Sehra[2]
Research scholar[1], Assistant Professor[2],
Department Of Computer Science and Engineering
Guru Nanak Dev Engineering College,
Punjab Techinical University, Ludhiana, 141006
Punjab - India

ABSTRACT

Several companies are planning to finance or have already financed money, time and resources in software reuse. Along this investment software companies expect to improve their completeness in the market through reduction of cost and effort, increase their productivity and improvement in the quality and reliability in the software products development.

Software reuse is considered as a key to a successful software development because of its potential to reduce the time to market, increase and reduce costs. This increase in demand made the software organization to envision the use of software recyclable resources which can also help in solving repeated problems. Till now, software reuse is confined to reuse of source code in the name of code searching. Now a day, software organizations are extending the concept to other life cycle objects as they realize that the reuse of source code alone does not save money.

In this thesis work the main focus is placed on developing a formula on the occurrence of true effort while developing a software project. In this work effort and cost has been calculated with reusable components of requirement analysis and specification phase, design phase, coding phase and testing phase. Because there is simple relationship between the development cost and the price charged from the customer and this is mainly because of the influence of economic, political and business consideration. We compare this work with Cocomo 81 and Cocomo Estimator dataset is taken as a source data for analysing the behaviour and pattern of effort associated with software development.

Keywords:- Software Reuse, Formula, COCOMO.

I. INTRODUCTION

Software reuse [3] is a process of developing a new software system using prewritten or existing software component. In the days of old, no or very little reusable components were present hence the software industry had to write entire software project from scratch. But in this day and age software industry has taken a giant leap towards developing software with incorporating reusable component primarily because:

i. **Component Abundance** - Large number of reusable component are found. It is very likely that for any software development project some or most reusable can be found easily.

ii. **Less Error –** Since reusable components are already tested and hence can be used without going through test/debug cycle.

iii. **Time Friendly –** Software production house can devote more time in creating more complex function without reinventing the wheel again.

iv. **Resource Saving-** Incorporating reusable component helps in utilizing necessary resources elsewhere otherwise which would be required to develop these components from scratch.

Cost of post deliverables were mere a guess work in the past because they had no similar project to serve as reference but now a days lot of project can be taken as guiding reference. Therefore, many methods and formula has been developed to estimate cost with high degree of accuracy.

Challenges with estimating cost of software when reusable component are introduced:

a) Extent of reusable component required is usually uncertain.

b) Though plethora of reusable component are present but searching them also incur cost and these cost are significantly increased with search-time.

c) Cost of tailoring reusable component to meet the current software development project is high.

d) Syncing reusable components with other component of software project is high.

Despite of these challenges introducing reusable component outweighs scratch development. Hence it of paramount need to deduce a metric or formula to accurately measure cost of software project beforehand but no such formula or method has been proposed in the literature. This paper bridges this gap by presenting few formulae based on mathematical model which may be used as a metric to estimate cost of software development cycle.

II. LITERATURE REVIEW

Hafedh Mili et.al., (1995)[4] discusses the effect of reuse on software production and its challenges. According to them locating and adapting those assets raises methodological issues and they classified software development cycle when reusable assets are being used:

1) The life cycle for developing reusable assets and
2) The life cycle for developing with reusable assets.

Additionally, they mooted the challenges in supporting reusability in software development are:

1) To identify reusable task
2) Provide methodology to support reusability and
3) Integrating reuse activities

They formulated the average cost of undertaking reuse as follow
S + (1-p) * D
Where,
S = Searching cost of reusable asset
P = Probability of finding the asset in database
D = Cost of developing the component from scratch

Kaushik et. al, (2013)[2] proposed a feed forward back propagation algorithm to train the neural network by iteratively processing set of input variable and adjusting the weights such that it can meet the predefined output. The outline of their algorithm are:

```
Step 1) Initialize the learning rate $\alpha$ and weights $w_i$
Step 2) Receive signal in input layer and send it to hidden layer
Step 3) Adjust weight on hidden layer
    a) $C_{EM} = b_1 + \sum z_i * x_i$
    b) $C_{SF} = b_2 + \sum z_i *(y_i + \ln(size))$
Step 4) Output layer calculate Cpm
Step 5) calculate error
Step 6) Update weight based on the calculated errors.
```

Mahajan et al, (2011)[1], proposed a model of reusability matrix using fuzzy logic for cost estimation for software project using reusable components. They classified software project into modules and according to them each modules can either be reused or developed from scratch. They characterized the modules further into three categories as

a) Completely reusable
b) Reusable with fragmentary adaptation and
c) Reusable with prominent adaptation

To implement the reusability matrix using fuzzy logic they implemented neural network and also developed software called REBEE that validates their model.

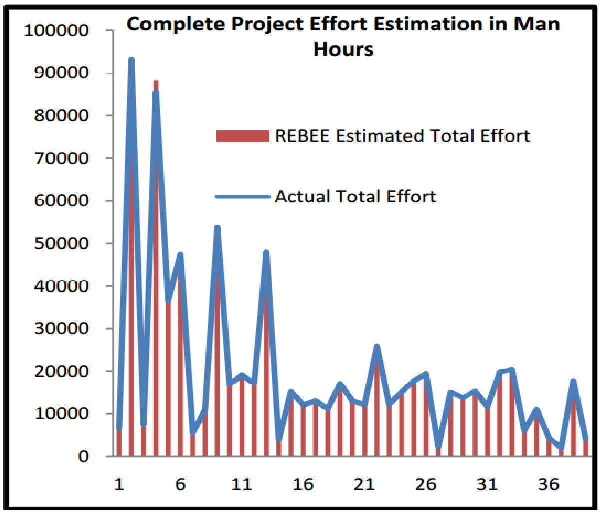

Figure 1: Effort Estimation using REBEE vs Actual Effort

The red line shows the prediction of the cost estimation vs the actual effort shown in blue.

III. METHODOLOGY

Estimates are made to discover the cost of producing a software system. Since there is no simple relationship between the development cost and the price charged to the customer and this is mainly because of the influence of economic, political and business consideration. But this thesis focuses on developing a formula that curtains the true effort incurred while developing a software project.

A. Analyzing Patterns

1) To analyze patterns of complete software development the patterns to be analyzed are broken down into sub-phase namely,
    a. Requirement Analysis and Specification Phase
    b. Design Phase
    c. Coding Phase and
    d. Testing Phase

Though software development team usually include cost of maintenance in their effort. But for this thesis this cost is not included as it is passive cost and usually generated after entire software project is over. And it is a long term cost from the release of the software to its span.

i. Requirement Analysis and Specification [5] Phase- Requirements analysis in software engineering circumscribes those functions that go into determining the needs or conditions to meet for a software project, taking account of the possibly conflicting requirements of the various stakeholders. In this phase task documents are generated analyzed and to ensure no conflict requirement raise upon completion of the project. Usually this phase start with verbal communication and dives deep into learning the atmosphere where this software is to be executed. This phase usually start with little to no effort and goes all the way to its maxima and then starts to fade up. To model this phenomenon mathematically even function is required.

Total Cost of Requirement Analysis and Specification Phase is the area under curve of project developed from scratch minus the area under curve of cost of incurring Reusable Components.

T(x) = C(E(x))-C(R(x))

$$T(x)=[A[\frac{a_n t^{n+1}}{n+1}+\frac{a_{n-1}t^{n}}{n}+\frac{a_{n-2}t^{n-1}}{n-1}\dots\frac{a_1 t^2}{2}+a_0 t]$$

$$e^{a_n t^n+a_{n-1}t^{n-1}+a_{n-2}t^{n-2}\dots a_1 t}+C]$$

$$-\frac{a((-g)^d g+(-g+bt)^{d+1})}{b+bd}+c$$

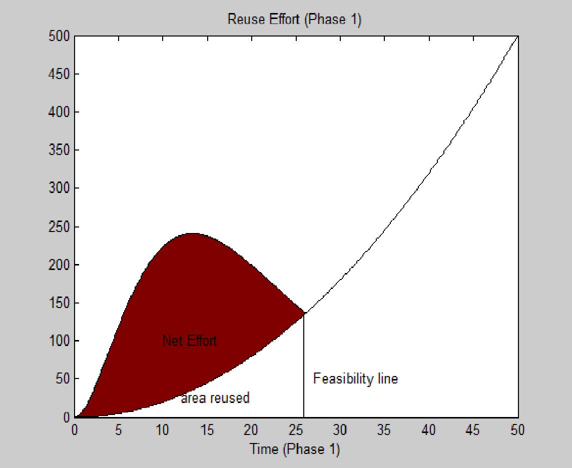

Figure 2: Total Cost of Requirement Analysis and Specification Phase

ii. The Cost of Design phase [6], Cost on introducing Reusable component [7] and Total cost of the code phase with Reusable component is given below:

Cost(x) = $\int_{t1}^{t2} a1 *(b_{1x}+g)^d + c_3$

Reusable(x) = $\int_{t1}^{t2} a2 *(b_2+g_2)^{(1/n)} + c_4$

Total(x) = $\int_{t1}^{t2} a1 *(b_{1x}+g)^d + c_3 - \int_{t1}^{t2} a2\,(b_2+g_2)^{(1/n)} + c_4$

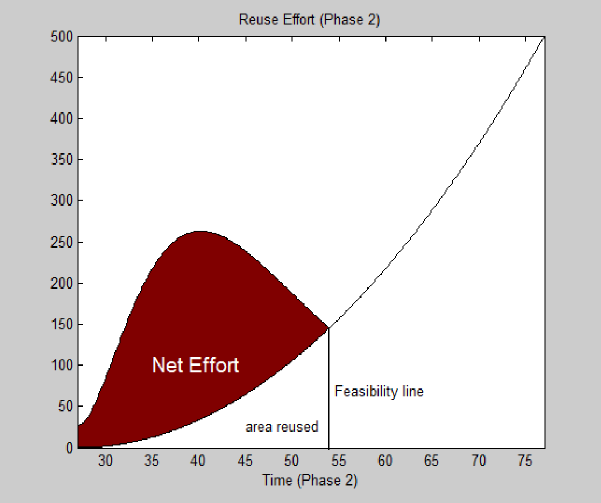

Figure 3: Total Cost of Design Phase

iii. The Cost of Coding phase[8], Cost on introducing Reusable component and Total cost of the code phase with Reusable component is given below:

Cost(x) = $\int_{t0}^{t1} k * (ax + g)^d + c_1$

Reusable(x) = $\int_{t0}^{t1} s * \sqrt{(bx+p)} + c_2$

Total(x) = $\int_{t0}^{t1} k *(ax+g)^d + c_1 - \int_{t0}^{t1} s *\sqrt{(bx+p)} + c_2$

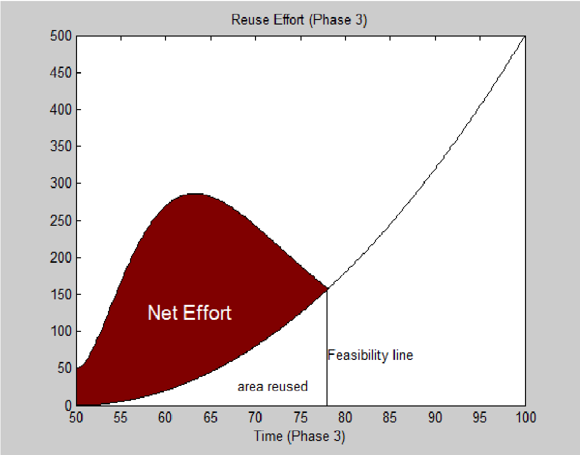

Figure 4: Total Cost of Coding Phase

The Cost of Testing phase [9], Cost on introducing

Reusable component and Total cost of the code phase with

Reusable component is given below:

$$Cost\,(x)=\int_{t2}^{t3} a_3\, x^{b_7}+c_5$$

$$Reusable\,(x)=\; c_6$$

$$Total(x)=\int_{t\,2}^{t\,3} a_3\, x^{b_7}+c_5-\,c_6$$

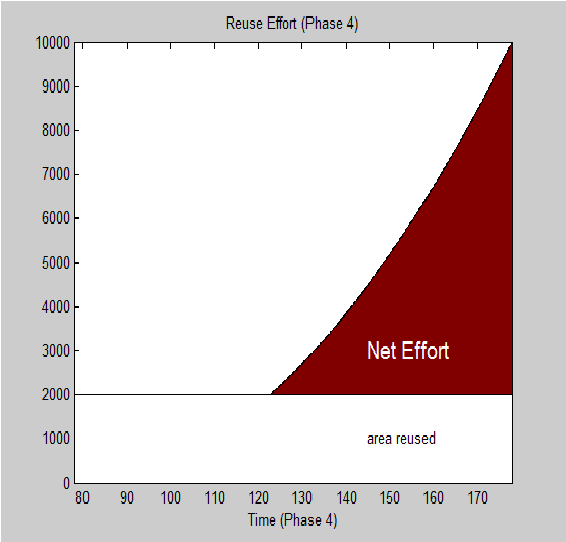

Figure 5: Total Cost of Testing Phase

B. Formula

Piecewise formula of different phases has been shown above. To obtain the complete formula each piecewise formula should be added together to estimate the complete cost incurred to software development team in creation of software development project.

$$\therefore \quad ProjectCost\,(x)=\int_{0}^{t_0} T_0+\int_{t_0}^{t_1} T_1+\int_{t_1}^{t_2} T_2+\int_{t_2}^{t_3} T_3$$

where,

T_0= *Cost for completion of Requirement Analysis* /\ *Specification Phase*

T_1= *Cost for completion of Design Phase*

T_2= *Cost for completion of Coding Phase*

T_3= *Cost for completion of Testing Phase*

IV. RESULTS AND COMPARISON

The proposed work is compared with COCOMO [10] model using COCOMO 81 dataset. Project number 1 is used for comparison.
Actual Effort of Project 1 = 2040
Lines of Code in developing Project = 113
Therefore, Computed Effort =
$2.4 * Kloc^{1.05}$=343.51334

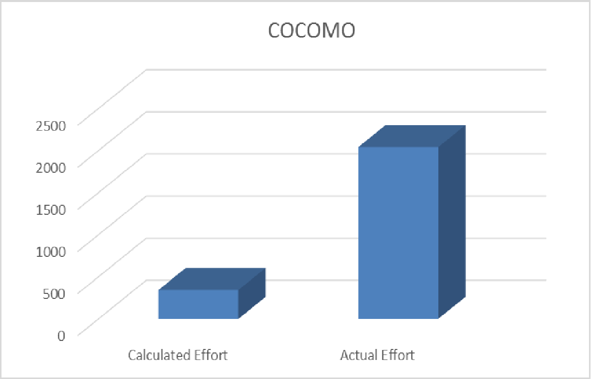

Figure 6: Precent Error COCOMO vs. Actual Effort

Percent Error = $\frac{|343.51334-2040|}{2040} * 100$ = 83.161108%
To evaluate the cost of proposed model the total project cost is passed through genetic algorithm and the derived cost of the project is given below:

Calculated Effort of the proposed formula = 2057.3345
Percent Error =$\frac{|2057.3345-2040|}{2040} * 100$ ≈ 1% (approx.)

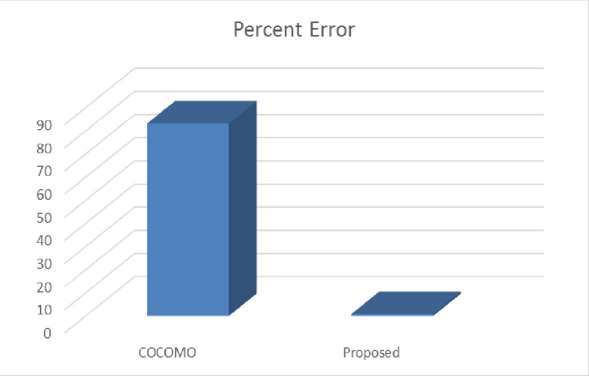

Figure 7: Precent Error of COCOMO vs. Proposed Formula

V. CONCLUSIONS

Accurate cost estimation of software project is crucial to any development house. Incorrect cost estimation can lead to project failure, loss of reputation and loss of client. Though this mathematical model for cost estimation has been tested with only one project. But it apparently demonstrates its strength in its versatility. This mathematical formula is highly customizable and can be incorporated with complete reusable module, and semi-complete reusable module projects.

VI. FUTURE SCOPE

Though this mathematical model gives us a generic formula as a starting point. Further neural network may be trained using this formula to give us exact values of the coefficient used in this formula. And this formula can be optimized further such that it can be implemented using a digital computer with $\Theta(n)$ running time and $\Theta(1)$ space complexity.

REFERENCES

[1]. *An Approach for Effort Estimation having Reusable Components in Software Development.* **Jyoti Mahajan and Devanand.** 20, s.l. : Global Journals Inc., December 2011, Global Journal of Computer Science and Technology, Vol. 11.

[2]. *A Simple Neural Network Approach toSoftware Cost Estimation.* **Anupama Kaushik, A.K. Soni and Rachna Soni.** 1, s.l. : Global Journals Inc., 2013, Global Journal of Computer Science and Technology, Vol. 13. 0975-4350.

[3]. *A pragmatic approach to software reuse.* **B. Jalender, Dr. A. Govardhan and Dr. P. Premchand.** 2006, Journal of Theoretical and Applied Information Technology.

[4]. *Reusing Software: Issues and Research Directions.* **Hafedh Mili, Fatma Mili and Ali Mili.** June 1995, IEEE Transactions on Software Engineering, Vol. 21. 0098-5589/95.

[5]. *An Optimization Approach to Analysis of Requirement Pre-Processing in Software Engineering.* **Chavan, Bechoo Lal and Dr. Chandrahauns R.** 2, February 2013, International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 3. 2277 128X.

[6]. *Understanding "Why" in Software Process Modelling, Analysis, and Design.* **Mylopoulos, Eric S. K. Yu and John.** Sorrento : s.n., 16 May 1994, Proceedings 16th International Conference Software Engineering.

[7]. *Building Reusable Software Component For Optimization Check in ABAP Coding.* **Sharma, P. Shireesha and Dr. S.S.V.N.** 3, s.l. : International Journal of Software Engineering and Application, July 2010, Vol. 1.

[8]. *Managing The Development of Large Software Systems.* **Royce, Dr. Winston W.** s.l. : The Institute of Electrical and Electronics Engineers Inc., August 1970, Proceedings of IEEE WESCON, pp. 1-9.

[9]. *Model Based Object Oriented Software Testing.* **Santosh Kumar Swain, Subhendu Kumar Pani and Durga Prasad Mohapatra.** Bhubansewar : s.n., Journal of Theroetical and Applied Information Techonolgy.

[10]. *Analysis of Software Cost Estimation using COCOMO II.* **Sharma, T. N.** 6, June 2011, International Journal of Scientific and Engineering Research, Vol. 2. 2229-5518.