



GLOBAL JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY

Volume 11 Issue 20 Version 1.0 December 2011

Type: Double Blind Peer Reviewed International Research Journal

Publisher: Global Journals Inc. (USA)

Online ISSN: 0975-4172 & Print ISSN: 0975-4350

An Approach for Effort Estimation having Reusable Components in Software Development

By Jyoti Mahajan, Devanand

University of Jammu, India

Abstract - Estimation of the effort required for development has been researched for over 25 years now. Still there exists no concrete solution to estimate the development effort. Prior experience in similar type of projects is a key for business today. This paper proposes an Effort Estimation Model named REBEE based on the reusable matrices to effectively estimate the effort to be involved for development. A project is assumed to consist of multiple modules and the reusability factor of each module is considered in the technique described here. REBEE utilizes fuzzy logic and dynamic neural networks to achieve its goal. Based on the experimental evaluation discussed in this paper it is evident that this model accurately predicts the effort involved on heterogeneous project types.

Keywords : *Software Effort Estimation, Software Reusability, Dynamic Neural Networks, Fuzzy Logic, REBEE.*

GJCST Classification : D.2.9



Strictly as per the compliance and regulations of:



An Approach for Effort Estimation having Reusable Components in Software Development

Jyoti Mahajan ^a, Devanand ^o

Abstract - Estimation of the effort required for development has been researched for over 25 years now. Still there exists no concrete solution to estimate the development effort. Prior experience in similar type of projects is a key for business today. This paper proposes an Effort Estimation Model named REBEE based on the reusable matrices to effectively estimate the effort to be involved for development. A project is assumed to consist of multiple modules and the reusability factor of each module is considered in the technique described here. REBEE utilizes fuzzy logic and dynamic neural networks to achieve its goal. Based on the experimental evaluation discussed in this paper it is evident that this model accurately predicts the effort involved on heterogeneous project types.

Keywords : Software Effort Estimation, Software Reusability, Dynamic Neural Networks, Fuzzy Logic, REBEE.

I. INTRODUCTION

SOFTWARE EFFORT ESTIMATION is crucial to derive the effort involved in the successful completion of any project. Effort estimation techniques facilitate financial estimates, delivery timelines, help in beneficial resource allocation and scheduling, monitoring progress and also help in risk management. According to a recent survey conducted by McKinsey for NASSCOM [1] the IT and allied industries are expected to bring in revenues of about \$225 Billion by 2020 in India alone and the current revenues are about \$76 Billion. It is evident from these figures the growth rate of the software industry is impressive. The recent years have observed that software contracts are awarded to organizations having prior experience in handling similar project types.

Prior experience in the related project is the key for business growth. Organization benefiting from the software contracts would have multiple reusable modules for their future work. More over organizations develop codes so that they could be reused with some modifications for future use. This conservative approach adopted by the industry is to ensure timely deliveries, quality, reliability and financial assurance of their investments.

Author ^a : Assistant Professor in Computer Engineering Department, Govt. College of Engineering & Technology, Jammu, India.
E-mail : jyoti_1972@sify.com.

Author ^o : Professor in Department of Computer Science & IT, University of Jammu, India. E-mail : padhadevanand@yahoo.com

COCOMO [3] and COCOMO 2.0[4], DELPHI [5], Function Point [6], Planning Poker [7], Use Case Point [8], Expert judgment [9], IBM – FSD [10] are the world known based estimation techniques, which are commonly used for Software Effort Estimation. These models exhibited a gross error of effort estimation. COCOMO with effort adjustment factor [11] provides about 30% improvement in effort variance, whereas when it is used with fuzzy logic, trapezoidal function and Gaussian functions showed improved performance [12]. Multiple software effort estimation techniques were integrated together to get the better result as compare to the regularly used estimation techniques, which was the big failure in terms of consistency when tested against several cases.

It was found that to achieve the good accuracy, Support Vector regression was combined with clustering approach. The estimation algorithm was vastly improved by the Mantel's correlation randomization test named Analogy-X [15]. This made the researchers to work even harder on the after effects of Schedule and Budget pressure on Effort Estimation and the development cycle time. Researchers have to be very careful while Chronological Splits are assigned for the testing and training purpose. Even Global Software Developments gets an inaccurate estimation technique being executed in different location of all over the world.

It has become very difficult to decide which model like COCOMO is best suitable for the development of the estimation model because of the different efforts to achieve estimation technique available in the market and the same outputs. The best solution for the estimation technique can be the judgment and the formal based model. In spite of all these available models and approaches, research shows the failure of projects due to various reasons [13]. Project Failures due to improper estimation techniques is also studied [14]. Based on this study it is evident that appropriate effort estimation techniques are critical for project success. The current existing techniques provide no proper estimation and are not applicable for varied project types.

To estimate effort for heterogeneous project types this paper discusses REBEE in the further sections of the paper. The remaining paper is organized as follows. The next section discusses the importance of

reusability and its adoption in the industry today. The third section discusses the REBEE model proposed. Section 3 also presents the Fuzzy rules to derive the reusability matrix and its use with dynamic neural networks to estimate effort. The penultimate section presents the experimental evaluation conducted using REBEE. The conclusion of the research presented here is discussed in the last section.

II. REUSABILITY AND ITS IMPORTANCE

The software industry today has witnessed various changes in its formulation, maintenance and management strategies to adapt to the dynamic changes it has experienced and for greater profitability. Experience held with organization in relevant or similar projects provides them with an business advantage as discussed earlier. These organizations possess modules which could be altered or used in total for their upcoming projects. The work described in this paper utilizes this knowledge of these reusable components to predict the effort required for the remaining work at hand. Incorporation and importance of reusability is currently been actively considered by major corporations now. Reusability is being considered for appraisals of employees of an organization [16] to reduce costs and maximizing profits [17]. Through these studies it is evident the adoption and importance of reusable components in the industry today and effort estimation using based on reusability could answer the anomalies that exist in the current estimation techniques adopted.

Fellow researchers have incorporated reusable weights into the existing COSYMO for cost estimation [18].Incorporation of the reusable parameters with the taguchi model [19], COCOMO2 [20],COCOMO [11] and COCOMO81[20] have been closely observed and these models exhibit considerable improvements but the error of estimation still exist. The error in estimation is basically due to the fact that the deficiencies of reusability's were not considered [21] which was considered to develop REBEE.

III. REBEE

a) REBEE Preliminaries

REusability Based Effort Estimation technique abbreviated as REBEE [2] consists of 4 phases as shown in Fig 1.

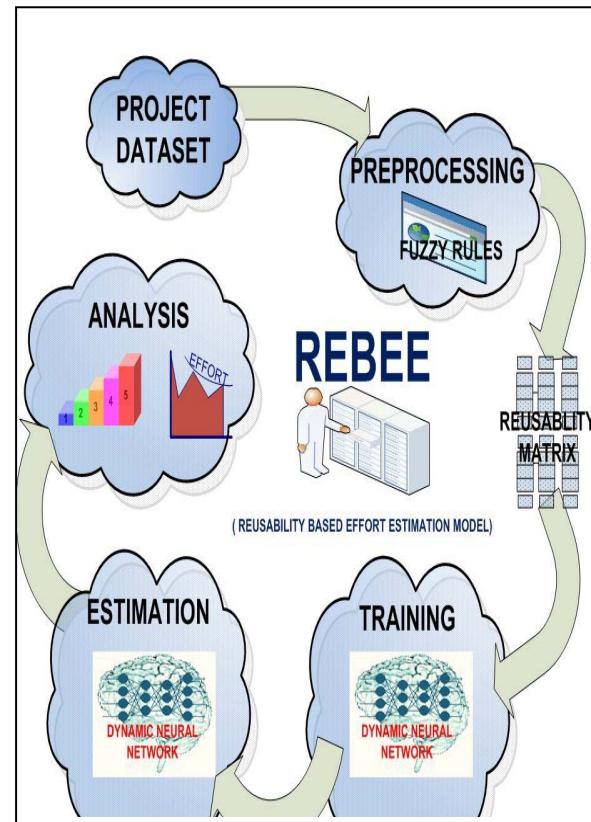


Figure 1 : REBEE Model

The effort estimation technique proposed consists of a pre processing phase where in the project data considered is analyzed to basically derive the reusability matrix. A project is assumed to be split into a number of modules and the reusability of each module is analyzed to derive the reusability matrix using fuzzy rules.

Estimation of the effort involved to achieve the project goals have been achieved using dynamic neural networks. Prior to estimation the dynamic neural networks are trained using the back propagation algorithm. The trained neural network could be used for estimation the effort involved. The results obtained could be analyzed for resource utilization, financial analysis, delivery time line assertion and many more critical analyses.

b) Reusability Matrix using Fuzzy Logic

A project is said to be composed of m modules. Modules could be either reusable or could be considered as new modules (m_n). Each reusable module is analyzed using a judgment model to arrive at the reusable component present. The modules are analyzed at an implementation level and for characterization a threshold ϑ is defined which is arrived based on the judgment model. On characterization the modules are further classified into 3 categories as

- Completely reusable.

A module is considered to be completely reusable if it could be utilized without any changes or

without altering any code or design and is represented as m_{cr}

- Reusable with fractional adaptation

A module is considered as a reusable model with partial adaption if at the implementation level the changes to be incorporated are less than the threshold ϑ and is represented as m_{fr}

- Reusable with prominent adaption

If the changes to be incorporated are greater than the threshold ϑ then the module is considered as a reusable module with prominent adaption represented by m_{pr} .

Let Δ represent the changes to be incorporated into a module m for it to be compatible with the project for which estimation is to be achieved. Applying the fuzzy rules the modules could be characterized as follows

$$\begin{cases} m = m_{cr} \text{ if and only if } \Delta = 0 \\ m = m_{fr} \text{ if and only if } \Delta < \vartheta \\ m = m_{pr} \text{ if and only if } \Delta \geq \vartheta \end{cases}$$

Consider R to represent the reusable matrix. The effort involved to develop the modules earlier is represented as \mathcal{E} . Let us consider that there exist x, y and z number of m_{cr}, m_{fr} and m_{pr} modules and their development efforts considered be defined as $\mathcal{E}_{cr}, \mathcal{E}_{fr}$ and \mathcal{E}_{pr} . Then the reusability matrix obtained based on fuzzy logic could be represented as

$$R = \begin{bmatrix} m_{cr1} & \mathcal{E}_{cr1} & m_{fr1} & \mathcal{E}_{fr1} & m_{pr1} & \mathcal{E}_{pr1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ m_{crx} & \mathcal{E}_{crx} & m_{fry} & \mathcal{E}_{fry} & m_{prz} & \mathcal{E}_{prz} \end{bmatrix}$$

c) Estimation using Dynamic Neural Networks

To estimate the effort involved REBEE uses neural networks. Neural networks of static type are not considered to develop this effort estimation technique as they possess adaptive and learning capabilities for only static in-out relationships. To effective adapt and learn the dynamic input output of the non linear matrices REBEE uses dynamic neural networks. The dynamic Neural Network Adopted in this model is as shown in Fig 2.

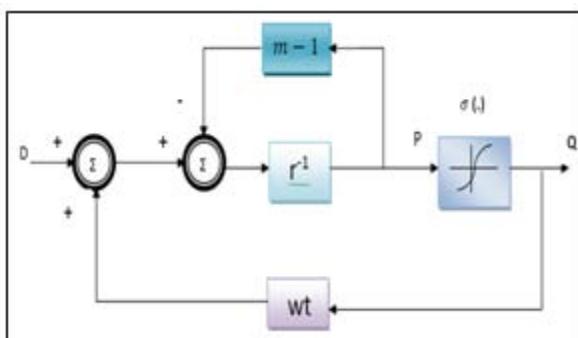


Figure 2: Dynamic Neural Network with Back Propagation Learning

Dynamic neural networks have been considered as they could be utilized to observe effort related dynamics of the input pattern matrices. The use of dynamic neural networks is not only related to obtaining effort related dynamics but also could be utilized to obtain non effort related dynamics observed for effort related input matrices.

The reusable matrix obtained from the pre processing phase is considered for training of the dynamic neural networks. The training is achieved using the back propagation algorithm. The output of the dynamic neural network $r(k)$ with respect to the input $p(k)$ is given by

$$\begin{aligned} p(k+1) &= -(m-1)p(k) + wtq(k) + d \\ r(k) &= \sigma(p(k)) \end{aligned}$$

Where σ represents the sigmoid activation function and $(m-1)$ is the feedback where m is the learning rate constant.

The error of estimation $E(k)$ is defined as

$$\begin{aligned} E(k) &= \frac{1}{2}(p_d(N) - p(N))^2 + \frac{1}{2} \sum_{k=0}^{N-1} [p_d(k) - p(k)]^2 \\ &= \frac{1}{2}e^2(N) + \frac{1}{2} \sum_{K=0}^{N-1} e^2(k) \end{aligned}$$

The weight update function $\Delta wt(k)$ propagated through the dynamic neural network is given as

$$\Delta wt(k) = -m \frac{\partial E}{\partial wt} = m \sum_{k=0}^{N-1} r(k+1) f_{wt}(p(k), wt)$$

The updated weights propagated to the next neuron based on the previous neuron is given as

$$\begin{aligned} wt(k+1) &= \{wt(k) \\ &- m_{wt} \sum_{k=0}^{N-1} r(k+1) f_{wt}(p(k), wt)\} \end{aligned}$$

The trained neural network is queried with the project data provided which provides the effort estimated on the remaining modules using the following equation where E represents the effort.

$$p(k+1) = -(mE - 1)p(k) + Ef(p(k), wt) + Ed(k)$$

This section of the paper described the REBEE technique proposed through this paper. The validation of this model is provided in the next section.

IV. EXPERIMENTAL VALIDATION OF THE REBEE TECHNIQUE

This section of the paper would discuss the experimental evaluation of the discussed REBEE model. For evaluation 39 projects of NASA Goddard Space Flight Center Greenbelt, Maryland is considered [22]. The dataset consists of projects related to simulators and altitude ground support systems developed by the Flight Dynamics Division of Goddard Space Flight Center situated in Maryland USA. The simulator projects considered were categorized into dynamic simulators and telemetry simulators. The 39 projects considered were said to be developed in 3 phases. Phase 1 consist of the design Phase. The coding was considered as the second phase and the last phase was the testing phase. For evaluation purpose the effort involved in providing support towards these projects developed was not considered.

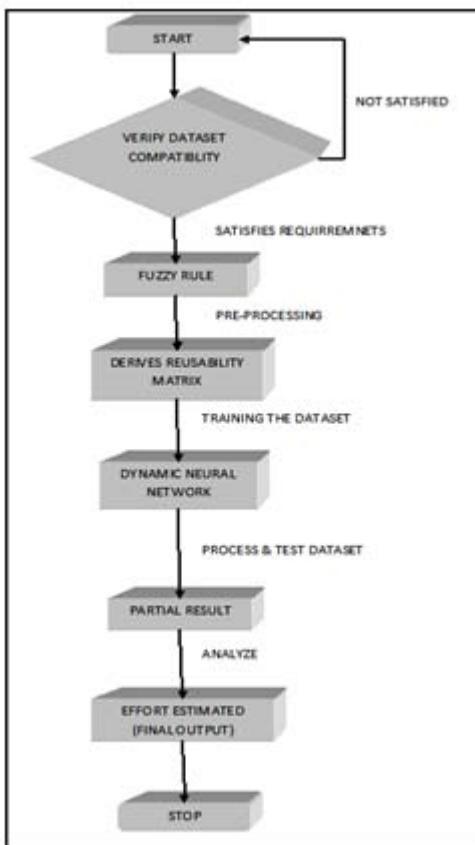


Figure 3 : Experimental Evaluation Flow Diagram

The data set considered provided details with respect to the number of lines of source code required in developing these projects. Reusability of the code was also considered in the development of the projects in the data set. The data set defined reusability of 3 types. A completely reusable code was considered if there were no changes to be incorporated for the new project considered. If the changes to be incorporated in

the code were less than 25% (i.e. threshold ϑ in REBEE) then it was considered to be a reusable code that requires slight modification. If the modification exceeded the threshold ϑ the code was considered to be reusable but with extensive modification. For evaluation presented here these matrices were considered to derive the reusability matrix R .

REBEE was developed using C# on a visual Microsoft Visual Studio 2010 platform. The reusability matrix derived using the fuzzy rule was provided to the dynamic neural network in the training process. The trained dynamic neural network on querying provides the effort estimated phase wise and for the entire project. The experimental evaluation process considered is shown as a flow chart in Fig 3. The dataset considered consists of heterogeneous project having varied development platforms and also exhibiting varying reusability levels. For evaluation projects were clustered into 4 types mentioned below

- Minor Reuse
- Standard Reuse
- High Reuse
- Maximum Reuse types.

If the reusability of a project was found to be less than or equal to 20% it was considered to be of Minor Reuse Type. If the reusability percentage of a project was between twenty and fifty, it was considered as a project of Standard Reuse type. If the percentage of reusability of a project was between fifty and eighty it was considered as a project of High Reuse. Projects embodying components which were more than 80% reusable was considered as maximum reusable projects. This clustering was adopted to provide for effective and efficient training to neural network to understand the dynamics of reusability. The effort estimated versus the actual effort involved in the design phase of the 39 projects is shown in Fig 4. The effort estimated using REBEE for the coding and the testing phase is shown in Fig 5 and Fig 6. The effort estimation for the entire project which includes the design, coding and the testing phase is as shown in Fig 7. The horizontal axis of the graphs represent the project ID and the effort in man hours id plotted on the vertical axis of the graph.

The results obtained from the evaluation of the 39 projects considered exhibited a low average error in effort estimation of about 1.25%. The average error in effort estimation for the design phase was 1.34%, 1.38% for the coding phase and 2.29% for the testing phase respectively. Based on the graphical data provided and the low average error of estimation it is evident that the reusability based effort estimation technique presented in this paper could be effectively utilized to estimate the effort involved in developing a project.

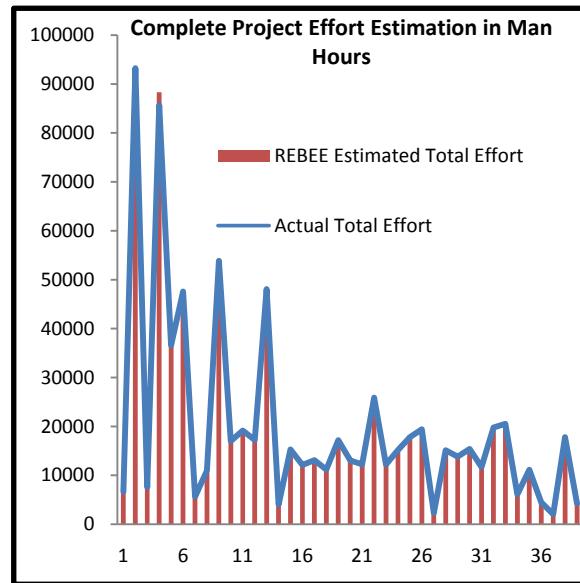


Figure 4 : Effort Estimation using REBEE Versus Actual Effort for 39 NASA Projects

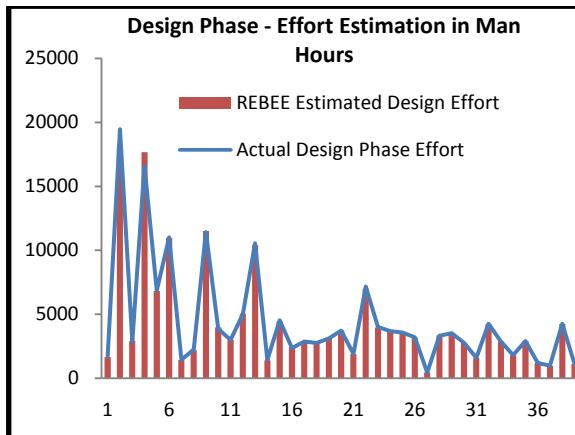


Figure 5 : Effort Estimation using REBEE Versus Actual Effort for the Design Phase

V. CONCLUDING REMARKS

Accurate effort estimation techniques are critical for the successful project execution. The importance of reusability and its remarkable acceptance by the industry today is evident from the research work presented through this paper. This paper discusses a reusability based effort estimation technique named REBEE.

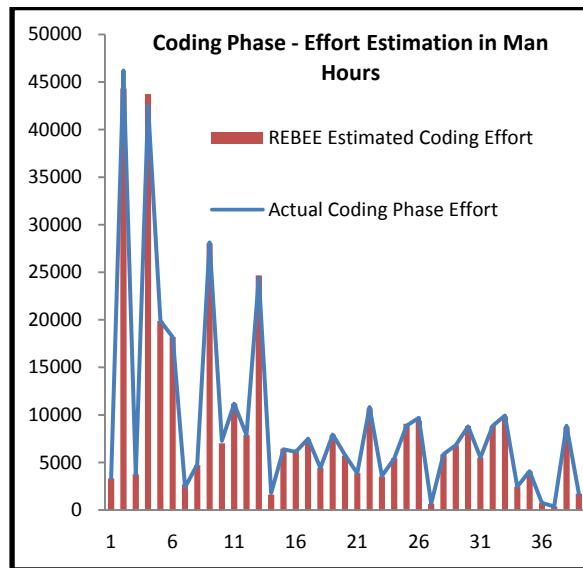


Figure 6 : Effort Estimation using REBEE Versus Actual Effort for the Coding Phase

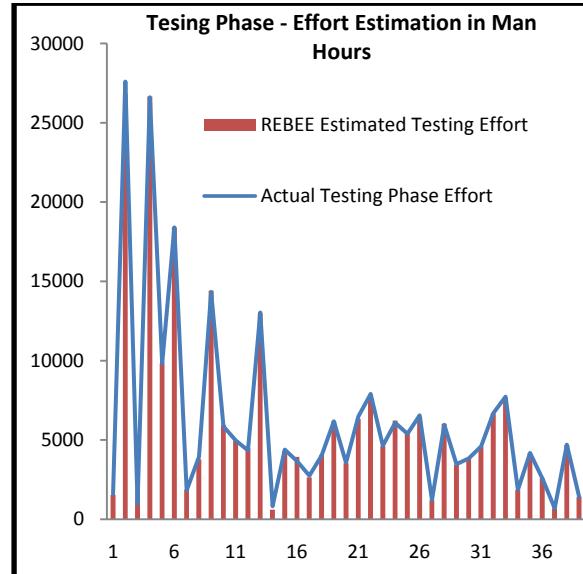


Figure 7 : Effort Estimation using REBEE Versus Actual Effort for the Testing Phase

REBEE achieves effective effort estimation utilizing the benefits of fuzzy logic and dynamic neural networks. Training of the dynamic neural networks is achieved using the back propagation algorithm. Fuzzy rules are adopted in constructing the reusability matrix which is utilized by the neural network to understand the dynamics of the effort involved in constructing the reusable components. Based on this understanding the dynamic neural network estimates the remaining effort involved in project completion.

The REBEE model discussed is evaluated on 39 NASA projects which are of different kinds. The development languages for these projects also varied from project to project. The reusability level of the

projects varied from about 0% to a high of 96%. The effort estimated using REBEE on all the 3 project phases i.e. Design, Coding and Testing and on the cumulative effort required in developing the projects showed high levels of accuracy. The average estimation error for all the 39 projects was also a low of about 1.25% which proves the efficiency of REBEE. From the evaluation results obtained it could be concluded that reusability based effort estimation technique discussed in this paper could be a possible solution for accurate effort estimation for projects of varied types which is not possible with the currently existing effort estimation techniques.

ACKNOWLEDGEMENT

The authors would like to express their cordial thanks to Prof. Bhopinder Singh, Principal of Government College of Engineering and Technology, Jammu for their financial support and advice.

REFERENCES REFERENCIAS

1. NASSCOM "Perspective 2020: Transform Business, Transform India" <http://www.nasscom.in/upload/Perspective%202020%20Press%20release%20presentation.pdf>
2. J. Mahajan, Devanand, K. Dhruve, "REBEE Reusability Based Effort Estimation Technique using Dynamic Neural Network", *Global Journal of Computer Science and Technology*, volume 11 Issue 7 ver 1.0 May 2011.
3. B.W. Boehm, W.W. Royce, Le COCOMO Ada, *Genie logiciel & Systemes experts*, 1989.
4. B.W. Boehm, et al., "Cost Models for Future Software Life Cycle Processes: COCOMO2.0", *Annals of Software Engineering on Software Process and Product Measurement*, Amsterdam, 1995.
5. Website:<http://www.stellman-greene.com/aspm/images/ch03.pdf>.
6. Meli, R., L. Santillo, "Function point estimation methods: a comparative overview", in Proc. 1999 *The European Software Measurement Conference - Amsterdam*, October 6-8.
7. Webreference : http://www.mountaingoatsoftware.com/system/presentation/file/51/bayXP_070320_PlanningAgileProjects.pdf
8. S. Nageswaran "Test effort estimation using use case points" in *14th International Internet Software Quality Week 2001, San Francisco, California, USA*, June 2001.
9. M. Jørgensen, "Practical Guidelines for Expert-Judgment- Based Software Effort Estimation," *IEEE Software*, vol. 22, pp. 57-63, May-June 2005.
10. C.E. Walston, A.P. Felix, "A method of programming measurement and estimation," *IBM Systems Journal*, vol. 16, no.1, 1997.
11. M.J. Basavaraj, K.C Shet, "Empirical validation of Software development effort multipliers of Intermediate COCOMO Model" *Journal of Software*, vol. 3, vo. 5, pp 65 MAY 2008.
12. C.S. Reddy, KVSN Raju, "An Improved Fuzzy Approach for COCOMO's Effort Estimation using Gaussian Membership Function". *Journal of Software*, vol. 4, no. 5, pp. 452-459, 2009.
13. C.E.L. Peixoto, J.L.N. Audy, R. Prikladnicki, "Effort Estimation in Global Software Development Projects: Preliminary Results from a Survey," in Proc. 2010 *5th IEEE International Conference on Global Software Engineering, ICGSE*, pp.123-127.
14. K. Molkken, M. Jorgensen, "A Review of Surveys on Software Effort Estimation," Proc. 2003 *International Symposium on Empirical Software Engineering (ISESE'03)*, pp. 223.
15. E. Kocaguneli, A. Tosun, A. Bener. "AI-Based Models for Software Effort Estimation" in Proc. *36th EUROMICRO Conference on Software Engineering and Advanced Applications*, pp.323-326.
16. P.S. Sandhu, H. Singh, "Automatic Reusability Appraisal of Software Components using Neuro-Fuzzy Approach", *International Journal of Information Technology*, vol. 3, no. 3, pp. 209-214, 2006.
17. P.S. Sandhu, H. Kaur, A. Singh, "Modeling of Reusability of Object Oriented Software System", *Journal of World Academy of Science, Engineering and Technology*, no. 56, pp 162 August 2009.
18. G. Wang, R. Valerdi, J. Fortune, "Reuse in Systems Engineering", *IEEE Systems Journal*, vol. 4, no. 3, pp 376-384, September 2010.
19. P. S. Sandhu, P. Blecharz, H. Singh, "A Taguchi Approach to Investigate Impact of Factors for Reusability of Software Components", *Journal of World Academy of Science, Engineering And Technology*, vol 25, pp 135-140, 2008.
20. CH.V.M.K.Hari, P.V.G.D.P Reddy, J.N.V.R.S Kumar, G.SriRamGanesh. CH.V.M.K. Hari., "Identifying the Importance of Software Reuse in COCOMO81, COCOMOII", *International Journal of Computer Science and Engineering JCSE*, vol. 1 no. 3, pp 142-147, 2009.
21. N. Ozarin, "Lessons Learned on Five Large-Scale System Developments" *IEEE Instrumentation & Measurement Magazine*, nol. 11, Issue- 1, pp 18-23, February 2008.
22. S. Condon, M. Regardie, M. Stark, etal., "Cost and Schedule Estimation Study Report", Goddard Space Flight Center Greenbelt, Maryland 20771. SEL-93-002 .pp 1- 119 November 1993.