Science Arena Publications
Specialty Journal of Electronic and Computer Sciences
Available online at www.sciarena.com
2016, Vol, 2 (4): 9-20

Presenting a Hybrid Method to Increase the Accuracy of Effort Estimation in Developing Software Products

Abouzar Doulatabadi Nejad[1], Seyyed Hedayat Tarighi Nejad[2], Sayyed Jalalaldin Gharibi Karyak[3]

[1]Faculty of Computer Engineering,Najafabad Branch,Islamic Azad University,Najafabad,Iran
Email: a_doulatabadi@yahoo.com
[2]Department of Computer Engineering Yasooj Branch, Islamic Azad University Yasooj, Iran
Email: heva.tarighi@yahoo.com
[3]Department of Computer Engineering Yasooj Branch, Islamic Azad University Yasooj, Iran
Email:Gharibi.jalal92@gmail.com

Abstract: *Vast studies have recently been conducted on increasing the accuracy of effort estimation, one of the most important one of which is COCOMO II. Another challenge in software development management is effort distribution among the phases of software development. In the proposed method, in order to estimate and distribute effort to develop software product, hybrid methods by the use of several effort estimation methods can increase the accuracy of estimation. In this research, by presenting a hybrid model of effort estimation in software development, attempts have been made to reduce deviations in effort distribution at each phase in software development and finally to increase the accuracy of effort estimation. The presented effort estimation method is based on a combination of fuzzy expert system and experimental effort estimation model COCOMO II. Having conducted the evaluations using NASA data set, level of increase of effort estimation in the presented method was investigated using MARE and VARE metrics. Data analysis indicates a reduction in the level of evaluation metrics and consequently an increase in the accuracy of effort estimation using the proposed method.*
Key terms: *software development management, effort estimation, effort distribution, fuzzy logic, COCOMO II model*

INTRODUCTION

One of the most important responsibilities of management in software development is project planning and timing. Using scientific and academic methods along with experimental methods can be a significant step in optimal management and consequently better quality of software product [1]. Various models such as experimental and expert models have been presented for effort estimation in software development and each one of them attempts to increase the accuracy of estimation. As an example, various studies have been conducted in the experimental effort estimation COCOMO II (Consecutive Cost Model) [2] to increase the accuracy of the model based on the use of uncertain techniques like fuzzy and neural networks. One of the important management responsibilities in software development after effort estimation is effort distribution estimated on each development phase. Results of effort estimation often include deviations and the deviation often leads to time and financial overheads in projects and at times leads to failure in the project. An example of effort distribution on each phase of software development is presented [3].

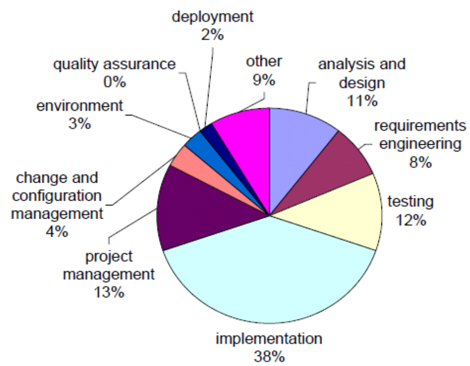

Figure 1: effort estimation in software development phases

Fewer studies on the causes of deviation in effort distribution than on effort estimation have been conducted and each of them have just studied the influential factors on effort estimation in software development and at some points models have been proposed with somehow less deviation in practice, one of which is [4]. The level of effort distribution in each software development phase is shown in the following figure based on software size [5].

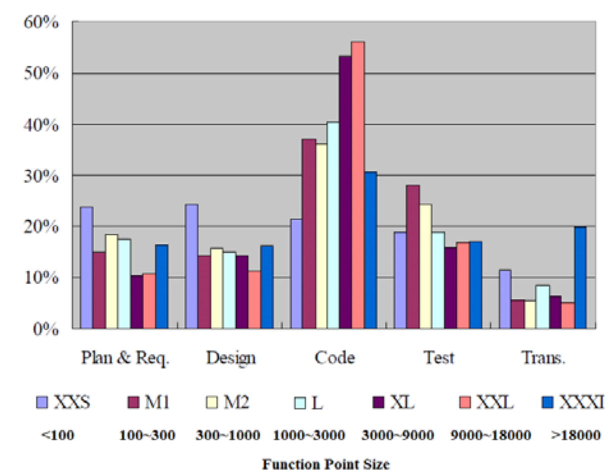

Figure 2: effort distribution in software development phases based on Software size

For example it can be seen in the above figure that the highest effort in software development in XXL (Extra Extra Large) size is dedicated to Code phase. Another influential factor in effort distribution is software domain factor. The level of effort distribution can be seen in each software development phase in the following figure based on software domain factor [6].

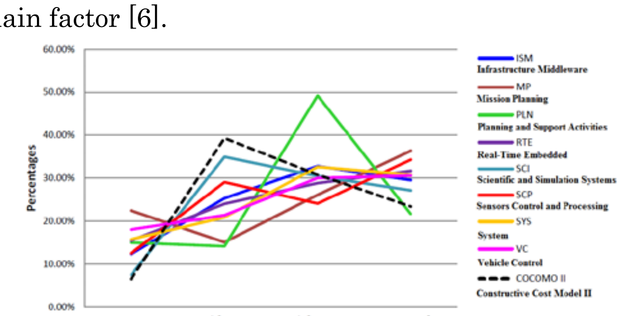

Figure 3: effort distribution in software development phases based on software domain factor

In effort distribution based on software development domain, the highest effort required in products with software domain of PIN (Planning and Support Activities) is for Unit Test and Code phase. Another important factor in effort distribution is development type (D_Type: Development Type) that can play an important role in deviation of effort distribution. Effort distribution based on software development type is studied in the following figure [5].

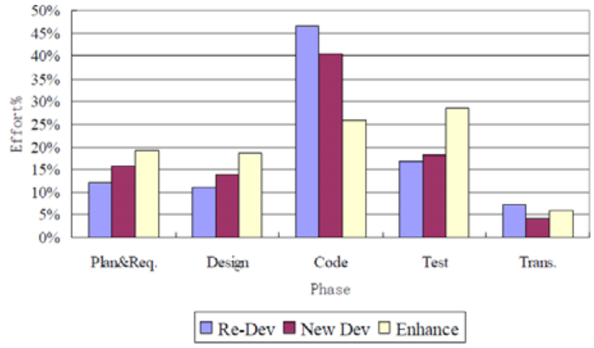

Figure 4: effort distribution in software development phases based on software development type

As an example, the highest effort is dedicated to Enhancement products in Test phase. The aforementioned points in the above figures can be a good guide for effort distribution by project management team. It is noteworthy that in previous studies, different data sets like COCOMO 81 [7] and ISBSG [8] data sets have been used to present methods or statistical studies and the presented models have confirmed the results on these databases. In the present research, by studying the previous research and extracting influential factors on deviation of effort distribution in software development, the level of deviation based on fuzzy logic has been investigated and finally attempts have been made to reduce deviation of effort distribution in software development by presenting a hybrid model. Fuzzy expert system and the experimental model COCOMO II are used in the proposed hybrid model to distribute effort with less deviation and finally effort estimation with high accuracy is obtained by regulating the required effort for each development phase. Evaluations made on NASA [11] data sets by the use of MARE (Mean Absolute Relative Error) [9] and VARE (Variance Absolute Relative Error) [10] metrics indicate an increase in effort estimation in software development using the proposed hybrid model to reduce deviation of effort distribution in software development. In the second part of this article, related works on this field will be dealt with and in the third part, the proposed hybrid model will be presented and described. The presented model will be evaluated in the fourth part and evaluation results will be analyzed in part five. Finally conclusions and suggestions will be made for further studies.

1. Related Literature

Different models have been presented to estimate effort estimation required for software development, one of the most important and famous one of which is COCOMO II which was first presented in 1988 by Bohem and is still one of the most reliable effort estimation models with the many changes made on it [12]. Different types of effort estimation models developed over the years have been classified in the following table.

Table 1: the most important effort estimation models in software development

Estimation approach	Examples of applied estimation approach
Comparative estimation	Angel, Weighted Micro Function Points
Estimation with failures (bottom up)	Project Management Software, Company Specific Activity Templates[13]
Parametric models	Cocomo [12], SlimSeer-Sem
Estimation models based on software size	Function Point Analysis[14],Use Case Analysis, Ssu

	(Software Size Unit), Story Points-Based Estimation In Agile Software Development
Group estimation	Planning Poker, Wideband Delphi[15]
Machine (mechanical) estimation	Average of An Analogy-Based And a Work Breakdown Structure-BasedEffort Estimate
Estimation by judgment	Expert Judgment Based on Estimates From A Parametric Model And Group Estimation

Some of the most important effort estimation models in software development in which an expert system is used have been presented in the following table which studies the newest models in effort estimation based on artificial intelligence and expert system.

Table 2: new models of effort estimation in software development using expert system

Researcher	Advantages and Disadvantage
Cherng Lin [16]	Presenting an expert effort estimation model based on Particle Swarm Oprimization Algorithm
Jyoti Mahajan [17]	Presenting COREAN model by the use of neural network learning
Roheet Bhatnagar [18]	Investigating effort estimation methods at the beginning of project expansion using fuzzy logic and neural network
S. Malathi [19]	Presenting a method based on machine learning to estimate software cost and effort

2. Description of the Proposed Method

In this section, the proposed method to increase the accuracy of effort estimation in software development is presented. In the first part, the hybrid effort estimation method is presented and described and in the second part, the fuzzy control design stages are investigated to be used in the hybrid effort estimation model.

3. Hybrid Effort Estimation Model

As it was previously mentioned, the purpose of using a combination of COCOMO II and fuzzy expert system in the presented method is for effort estimation. Since a combination of two models has been used for effort estimation, it is called a hybrid model. Effort is estimated based on this hybrid model in the proposed model by calculation of the level of the effect of influential factors on deviation of effort distribution and using the designed fuzzy control and application of the level of deviation for each influential factor such as development domain, software size and development type in Drive Distribute format that is introduced as the level of effect on effort distribution. An overview of the steps for effort estimation by the proposed method is presented in the following figure.

In model

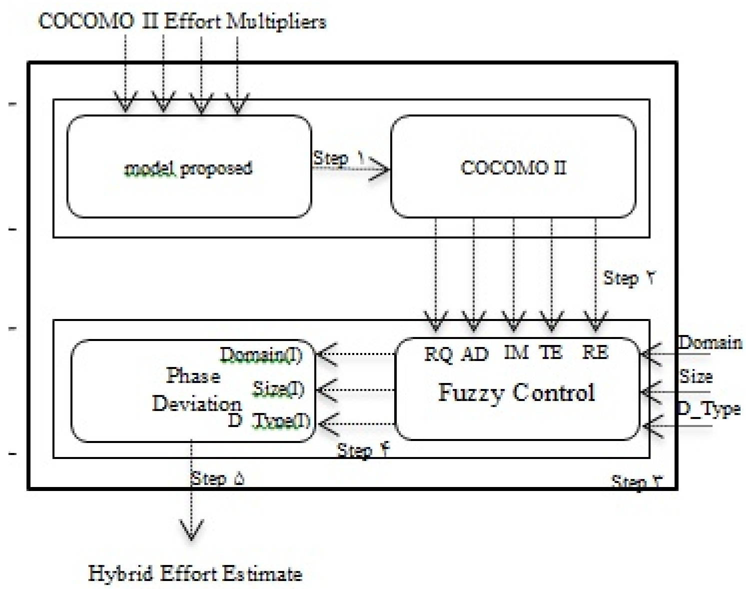

Figure 5: overview of the proposed effort estimation model

The hybrid model proposed to estimate effort that can be seen in the above figure is made of two expert and experimental sections. First the initial effort based on the experimental model will be estimated at first using the required inputs for effort estimation and COCOMO II which is shown as stage (1) in the above figure. Then, using the experimental effort distribution model of COCOMO II, effort will be distributed at each phase of software development (RE, TE, IM, AD, and RQ) which is shown as stage (2) in the figure. For a better understanding of the proposed model, the applied phase classes are presented in the following table for software development.

Table 3: research phase classification for software development

Phase	Activities
Requirement (RQ)	Requirement Analysis
Design(AD)	Product Design, Detailed Design
Code(IM)	Code, Unit test, Integration
Test(TE)	System test
Transition(RE)	Installation, Transition, Acceptance Test, User Training, Support

At stage (3), the three factors Domain, Size and Type-D are studied using the tables presented in the next parts. Then along with the results from stage (2), the designed fuzzy control is presented as the input of fuzzy control. The designed fuzzy control whose design will be investigated in details in the next part attempts to determine the effect of influential factors on deviation of effort distribution at each phase of software development at stage (4) by the use of the designed deductive rules, variables and linguistic terms presented in its design and fuzzy deduction.

In the final stage, drive distribute should be calculated to estimate effort. It can be achieved by multiplying all results from fuzzy control. For example, for RQ phase, the following formula is used to calculate drive distribute.

(1)

Having calculated Drive Distribute for all available phases, the level of deviation of effort distribution will be calculated which is influential in the accuracy of effort estimation. The following procedure will be taken.
First the required effort percentage for all software development phases will be calculated which is:
(2)
2) In the following, by the use of the subtraction between the obtained total efforts from previous stage, the level of deviation of effort estimation in COCOMO II will be calculated by the effort estimation COCOMO II. In the following formula, A variable is used as the deviation percentage of effort distribution in COCOMO II and B and C variables are used as effort estimations in the hybrid model and COCOMO II.
(3) B-C
3) Finally, by adding the obtained value from the deviation of effort distribution in the previous stage to the estimated value by COCOMO II, the final estimation will be achieved by the hybrid model which can be seen in the following.
(4)
As it was seen in all previous stages, in this method first by estimating effort by the use of the official model COCOMO II and distributing the effort based on the experimental effort distribution model COCOMO II, the level of influential factors of Domain, Size and D-Type on deviation of effort distribution was investigated and then by designing a fuzzy control, the level of this effect was studied and evaluated based on an expert system. Finally, by applying Drive Distribute which is the sum of the effect of influential factors on effort distribution in effect distribution model COCOMO II, the level of the newly estimated effort was studied and calculated based on the hybrid model. Results of the evaluations indicate an increase in the accuracy of effort estimation in the proposed method. In the next part, the fuzzy control design stages will be described in details to be used in the presented hybrid model in the effort estimation.

4. Design of the Proposed Fuzzy Control

In order to design the fuzzy control used in the proposed hybrid model, first the input and output linguistic variables and terms were extracted for each of the linguistic variables. Finally, the level of the effect of each one of the factors presented in the previous part was studied and required rules to use the fuzzy control were designed. The used input and output linguistic variables, linguistic terms of each one of the variables, fuzzy functions used in the variables and the tables used to extract rules required for fuzzy control can be seen in the classification presented in the following table.

Table 4: parameters of the proposed fuzzy control

Linguistic variables	Number of membership functions	Number of rules	Defuzzification	Type of membership functions	Type of designed control
Software size	5	25	Centroied (MAX-MIN)	trimf	Mamdani[20]
Development domain	5			trimf	
Development type	5			trimf	

All the input and output linguistic variables have linguistic terms as follows. As an example, the following linguistic term can be seen for the input variable software size.
Domain} =V-Low, Low, Medium, Large, V-Large{
In the following figure, all the linguistic variables used in the design of the fuzzy control can be seen.
Table 5: linguistic variables used in the designed fuzzy control

Input linguistic variables	Output linguistic variables
Domain	Domain(I)
Size	Size(I)
D-Type	D-Type(I)

2) All the fuzzy functions are trigonometric functions [21]. For example, the fuzzy function used in the following figure can be seen for domain input variable.

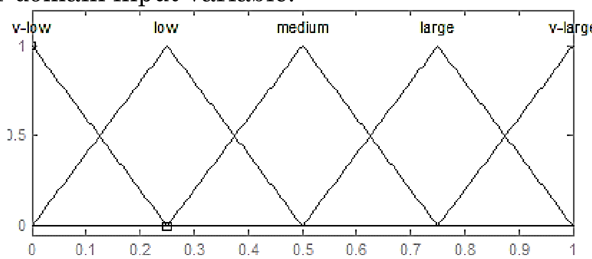

Figure 6: fuzzy functions used for domain linguistic variable in Matlab

3) The following tables were designed and extracted to extract the required rules in the designed fuzzy control by the use of the previous research on the level of the effect of domain [6], size [5] and D-Type [5] factors. Tables of Size, D-Type and Domain factors can be seen in the following.

Table 6: investigation of the level of the effect of Size factor on deviation of effort distribution to extract fuzzy rules [5]

Size	RA	AD	IM	TE	RE
XXS	7.72	9.42	18.88	2.56-	4.30
XS	1.18-	0.66-	3.29-	6.55	1.42-
M1	2.40	0.74	4.25-	2.74	1.63-
M2	1.47	0.01-	0.08-	2.64-	1.27
L	5.90-	0.55-	12.38	5.69-	0.69
XL	5.32-	3.66-	15.74	4.87-	1.89-
XXL	0.63	1.21	9.69-	4.56-	12.87

Table 7: investigation of the level of the effect of Domain factor on deviation of effort distribution to extract fuzzy rules [6]

Domain	RA	AD	IM	TE &RE
Business	20.98	22.55	24.96	31.51
Command & Control	21.04	22.56	33.73	22.66
Communications	14.95	30.88	28.54	25.62
Control & Display	14.72	34.80	24.39	26.09
Mission Management	15.40	17.78	28.63	38.20
Mission Planning	17.63	12.54	44.32	25.60
Sensors Control	7.78	45.7	22.2	24.19

and Processing		4	9	
Simulation	10.71	39.11	30.80	19.38
Spacecraft Bus	33.04	20.66	30	16.30
Weapons Delivery and Control	11.50	17.39	29.82	41.29

Table 8: investigation of the level of the effect of D-Type factor on deviation of effort distribution to extract fuzzy rules [5]

D_Type	RA	AD	IM	TE	RE
Redevelopment	3.64-	3.96-	8.95	-4.96	3.59
Enhancement	0.44-	0.12-	2.20	-1.07	-0.58
New development	2.82	1.88	-10.67	5.40	0.56

The required rules for the fuzzy control were trained and regulated by the data set present in the above tables using the neuro-fuzzy optimization method [22]. Results of trainings and evaluations of the proposed fuzzy control and the characteristics of the trained and evaluated data can be seen in figures 3 and 4.

Figure 7: coordinates of training data for the fuzzy control design

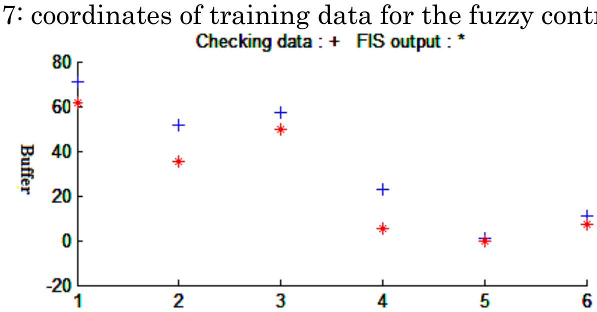

Figure 8: coordinates of evaluated data for the fuzzy control design

An overview of the fuzzy control designed in MATLAB [23] can be seen in the following figure.

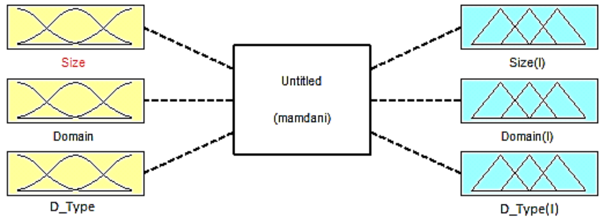

Figure 9: overview of the proposed fuzzy control simulation

5. Evaluation of the Proposed Method

Using NASA data set to investigate the accuracy of effort estimation, the presented hybrid model for effort estimation is evaluated in this part. A description of the data sets used in this evaluation along with the results and analysis of the evaluations can be seen in the next sections.

4.1. Data Sets Used in the Evaluation

NASA data set was used in this article to evaluate the presented method. The proposed method was applied on the projects presented in the following table and the obtained results for the evaluation of the proposed method can be seen in the next section.

Table 9: A part of the data set used for the evaluation of the proposed method

Effort Multiplayer	P21	P45	P47	P54	P62	P93
Rely (Required Software Reliability)	h	h	n	h	h	H
data(Database size)	l	h	n	n	n	n
cplx(Product complexity)	h	n	n	h	xh	vh
time(Execution time constraint)	n	n	n	h	h	vh
stor(Main storage constraint)	n	n	n	h	h	vh
Virt (Virtual machine volatility)	l	l	l	l	l	h
turn(Computer turnaround time)	l	l	n	h	l	h
acap(Analyst capability)	n	n	h	n	h	n
aexp(Applications experience)	n	h	vh	h	n	n
pcap(Programmer capability)	n	h	vh	n	n	n
vexp(Virtual machine experience)	n	n	l	n	h	l
lexp(Programming language experience)	h	h	h	n	h	l
modp(Use of modern programming practices)	h	n	h	l	h	n
tool(Use of software tools)	n	n	n	vh	h	n
sced(Required software development experience)	l	n	n	n	n	h
loc(Line of Code)	19.7	79	190	219	271	3
Actual Effort(PM)	60	400	420	2120	2460	38

Each multiplier effort in the above table will have the values: l: low, vl: very low, h: high, vh: very high, xh: extra high and n: nominal.

4.2. Evaluation Results

Having investigated the hybrid method proposed in the present study, results of the evaluation are extracted and the level of effort is estimated in the projects under evaluation. In the following table and diagram, the level of effort estimation in the presented method of each project is presented in comparison to the effort estimation model COCOMO II.

Table 10: comparison of effort estimation in the proposed model and effort estimation model COCOMO II in data sets under evaluation

ID	Actual Effort (PM: Person/Month)	COCOMO Method(PM)	Hybrid Method(PM)
21	60	50.6	80.99
45	400	400	483.11
46	2400	2400	3212
47	420	436.9	483.07
53	750	703.06	743.06
54	2110	2120	1137.81
62	2468	2356	2164
75	600	600	586.92
82	480	478.01	446.2
83	599	632.16	655.12
85	4148.2	8432.62	4029.18
86	1772.5	1239.6	1625.30
87	1645.5	1546.2	986.2
88	1924.5	1627.18	1890.32
93	38	31.9	25.36

Figure 10: comparison of effort estimation in the hybrid model and effort estimation model COCOMO II in data sets under evaluation

In the following and the next section, the obtained results in this section will be analyzed and the level of accuracy in effort estimation will be studied using the proposed hybrid method in this article.

Results Analysis

Among the common metrics for estimation of the accuracy of the effort estimation methods in previous studies such as [24] and [25], MARE and VARE metrics are used in this article to evaluate the proposed method. The following formulas are used to calculate MARE and VARE.

(5)

(6)

As it can be seen in the following table and diagram, the extracted results by the evaluation of effort estimation using the proposed hybrid method, it indicates a reduction in evaluative metrics in the proposed model and consequently an increase in the accuracy of the presented method for effort estimation compared to COCOMO II.

Table 11: comparison of MARE and VARE in the proposed method and COCOMO II

Estimation Model	MARE	VARE
Hybrid Model	23.85	32.65
II COCOMO	47.22	47.22

Figure 11: comparison between MARE and VARE for the proposed method and COCOMO II

By investigating MARE and VARE for evaluation of the accuracy of effort estimation by the proposed method compared to COCOMO II, it was found that the accuracy of the effort estimation has increased due to the reduction in deviation of effort estimation using fuzzy logic. As it can be seen in the above table and diagram, by the use of the proposed hybrid model in this study for evaluation data set, it has decreased 23.7 percent in MARE metric and 14.57 percent in VARE metric. The reduction in MARE and VARE in the proposed method means an increase in the accuracy of the proposed estimation compared to the effort estimation method COCOMO II in evaluation data set. The accuracy of effort estimation can be an important step in the success of software projects at management and executive levels and prevent from time and financial overheads in the project.

6. Conclusion and Suggestions

By investigating the methods and models of effort estimation and distribution in software development and considering the influential factors in the deviation of effort distribution, a hybrid model including the official COCOMO II along with a fuzzy control in effort distribution was presented in this study and attempts have been made to reduce deviation of the effort distribution in software development. This reduction in deviation finally leads to an increase in the accuracy of effort estimation. Having conducted evaluations on NASA data sets in this article, the results were presented in tables and diagrams in previous sections which indicate an increase in the accuracy of effort estimation in the proposed hybrid method in this model compared to COCOMO II. MARE and VARE accuracy metrics are used in the proposed model in this article to evaluate the accuracy of effort estimation. These metric are the most practical metrics in the field of effort estimation. One of the most important suggestions of this article for further studies is investigation of other Distribute Drives that influence effort estimation and the increase in its deviation. Additionally, using other models such as regression models or neural network models to investigate the effect of influential factors on deviation of effort estimation can be also suggested for further studies.

References

[1] S.R. Schach, *Object Oriented &Classical Sofiware Engineering.* Mc Graw Hill, New York, pp. 254-266, 2007.

[2] B.W. Boehm, C. Abts, A. Brown, C. Chulani and K. Horowitz, *Software Cost Estimation With COCOMO*

II. Englewood Cliffs, Prentice-Hall, pp. 0-13, 2000.
[3] W. Heijstek, "Empirical Investigations in Rational Unified Process Effort Distribution in Industrial Software Engineering Projects," Master's dissertation, Leiden Institute of Advanced Computer Science., Leiden Univ., Niels Bohrweg, Netherlands., 2007.
[4] T. Tan and B. Clark, "Technical Report of a New Taxonomy for Software Application Domains and Operating Environments," USC, CSSE Technical Reports, Center for Systems and Software Engineering., California, United States., 2011.
[5] Y. Yang, H. Mei, L. Mingshu, Q. Wang and B.W. Boehm, "Phase Distribution of Software Development Effort," *Proc. IEEE Symp. Empirical software engineering and measurement*, pp. 61-69, 2008, doi: 10.1145/1414004.1414016.
[6] T. Tan, "Domain-Based Effort Distribution Model For Software Cost Estimation," PhD dissertation, Dept. of Computer Eng., Southern California Univ., United States., Southern California., 2012.
[7] T. Menzies, "COCOMO 81Dataset," *promise,* http://promise.site.uottawa.ca/SERepository. 2004.
[8] P. Hill, *Practical Project Estimation.* McGraw-Hill, ISBSG., 2nd Edition, 2005.
[9] W. L. Du, D. Ho and L. F. Capretz, "Improving Software Effort Estimation Using Neuro-Fuzzy Model with SEER-SEM," *Global Journal of Computer Science and Technology*, vol. 10, no. 12, pp. 52-64, Oct.2010.
[10] P.V.G.D. Reddy, K.R. Sudha, P.R. Sree, "Application of Fuzzy Logic Approach to Software Effort Estimation," *International Journal of Advanced Computer Science and Applications*,vol. 2, no 5, 2011.
[11] T. Menzies, "COCOMO & NASA V1 Dataset," *promise,* http://promise.site.uottowa.ca/SERepositary/datasets/cocomonasa.arff. 2010.
[12] B. Boehm, A.W. Brown, and C. Fakharzadeh, "MBASE/RUP Phase And Activity Distributions (Presentation at COCOMO)," International Forum 14 Technical Report TR-90089, University of Southern California., Los Angeles, CA., 1988.
[13] A. Booz, G. Allen, V. Hamilton, "Earned Value Management Tutorial Module Work Breakdown Structure, Office Of Science", *Tools & Resources For Project Management*, Science, Energy.Gov, 2011.
[14] D. Longstreet, "Fundamentals Of Function Point Analysis", *Software Development Magazine*, doi: 2207 S-Mo 64015, 2005.
[15] A. Stellman, J.Greene, *Applied Software Project Management Sebastopol*, O'reilly Media, 2005.
[16] Y. T. L. Jin, H. Y. Tzeng and Y. C. Wang, " Using Computing Intelligence Techniques To Estimate Software Effort", *International Journal of Software Engineering & Applications (IJSEA),* vol: 4, no: 1, doi: 10.5121/ijsea.2013.4104, January 2013.
[17] S. D. Jyoti Mahajan, "COREAN: A proposed Model for Predicting Effort Estimation having Reuse,", *International Journal of Soft Computing and Engineering (IJSCE)*, vol: 2, no: 6, ISSN: 2231-2307, January 2013.
[18] M. K. Ghose, "Comparing Soft Computing Techniques For Early Stage Software Development Effort Estimations," *International Journal of Software Engineering & Applications (IJSEA)*, vol: 3, no: 2, DOI: 10.5121/ijsea.2012.3210, March 2012.
[19] S. Malathi, A. S. Sridhar, "Efficient estimation of effort using machine-learning technique for software cost," *Indian Journal of Science and Technology*, vol: 5, no: 8, ISSN: 0974-6846, August 2012.
[20] E. H. Mamdani, "Application of fuzzy algorithms for the control of a simple dynamic plant," *In Proc IEEE* , vol. 121, no. 12, pp. 121-159, doi: 10.1049/piee.1974.0328, December 1974.
[21] S. Kad and V. Chopra, "Software Development Effort Estimation using Soft Computing," *IEEE Proceedings of 3rd International Conference on Machine Learning and Computing (ICMLC '11)*, vol. 5, pp. 205-208, 2011.
[22] J.S.R. Jang, "ANFIS: Adaptive-Network-Based Fuzzy Inference System," *IEEE Trans. Systems, Man, Cybernetics*, vol. 23, no 5, pp. 665-685, 1993.
[23] Copyright 1995 - 2002 by The Math Works, Inc." Fuzzy Logic Toolbox For Use with MATLAB".
[24] I. Attarzadeh and S. Hock Ow, "Improving Estimation Accuracy of the COCOMO II Using an Adaptive Fuzzy Logic Model," *IEEE International Conference on Fuzzy Systems*, pp. 27-30, June 2011.
[25] B.A. Kitchenham, S.G. MacDonell, L.M. Pickard, M.J. Shepperd, "What Accuracy Statistics Really Measure," *IEEE Proceedings Software*, vol. 148, no. 3, pp. 81-85. 2001.