Tools & Methods for Software Effort Estimation Using Use Case Points Model – A Review

Meenakshi Saroha
M. Tech scholar, Computer Science and Engineering
Ajay Kumar Garg Engineering College
Ghaziabad, India
mnxi335@gmail.com

Shashank Sahu
Associate Professor, Department of Computer Science and Engg.
Ajay Kumar Garg Engineering College
Ghaziabad, India
sahu_shashank@yahoo.com

***Abstract* – Software effort estimation is primary requisite in software development life cycle. Most of the software projects failed due to inaccurate effort estimation. So, to overcome this shortcoming many techniques were introduced in past by various researchers. There are many techniques exists for estimating the software project effort such as learning oriented, model based and expert based techniques. This paper represents review on various techniques used for effort estimation but main focus is on the tools and frameworks developed for efforts estimation which based on Use Case Point (UCP) model. These tools come up with extra features such as consider more factors that may affect project delivery and ability to give better estimate than existing one.**

Keywords – Software effort estimation, use case point method, Tools

I. INTRODUCTION

Estimation plays crucial role in our software development process. Software effort estimation is a time consuming process in large software projects it will may takes around 20 to 40 percentage of the total time in requirement gathering which is the initial phase of software development life cycle. But if project effort is overestimated or underestimated it may affect the project delivery. So there is a need to calculate more accurate effort to avoid overestimation & underestimation problem. Therefore, to estimate effort various techniques were proposed by different researches.

According to [17], software estimation techniques classified into six parts out of which three techniques are popular. These are:

- **Algorithmic methods:** These methods based on mathematical models for the estimation of projects. COCOMO & COCOMO II and Putnam's model & SLIM are examples of Algorithmic model. SLIM tool is widely accepted and depend on this model for man power scheduling & estimation purpose.
- **Expertise based estimation:** Expertise based estimation is used when there is limitation of requirement gathering and in data finding. There are two types of this estimation method Delphi which uses combination of different expert opinion and Rule Based Systems in which expert knowledge is represented in the form of rules.
- **Learning oriented techniques:** These types of techniques uses historical data and current knowledge to develop model for software effort estimation. Analogy and neural network estimation are the examples of this type of categorization. Former one is based on the training and learning phase and later one is based on expert estimation and comparison with the results obtained from the historical data of the similar software project.

Some authors [7], [25], and [35] has classified the software estimation processes into six type of models:

- **Model based technique:** Model based techniques are helpful in planning, tradeoff analysis, controlling and budgeting. It consists SLIM, COCOMO, Check point, SEER methods. These techniques are not used nowadays for software effort estimation.
- **Expert based technique:** In the absence of empirical, quantified data expert based techniques are used. These techniques utilized the experience and knowledge of particular domain experts. Delphi, Work Breakdown Structure and rule based are the examples of this type of estimation technique. The main disadvantage of these types of techniques is that expert judgment may be biased or incorrect.
- **Learning oriented technique:** Learning - oriented techniques consist of both oldest as well as newest techniques used for estimation purpose. Former one represented by case studies while later one is represented by neural networks. These types of techniques require historical data in bulk. Case studies are manual techniques which provide result by examine specific example in detail. Neural network is one of the latest techniques which learn from experience or historical data and produce better results by automatically adjusting their algorithmic parameters.

ISBN:978-1-4799-8890-7/15/$31.00 ©2015 IEEE

- **Dynamics-based technique:** Dynamics-based techniques provide the acknowledgement that shows the change in software project cost or effort factors i.e. staffing level, deadline, and design requirements, training needs and budget during the software or system development. Abdul Hamid Madnick is one of the techniques involved in this approach. These types of techniques may be good for planning and controlling but difficult to calibrate.
- **Regression-based technique:** Regression based techniques are most famous technique for building estimation models. These are simple and widely accepted technique and used by some other models like COCOMO II, SLIM, checkpoint etc. Regression based techniques used in conjunction with model based technique. OLS (Ordinary Least Squares) and robust techniques come under this categorization. . However, it requires large amount of data, no data items should be missing.
- **Composite technique:** Composite techniques are the combination of two or more techniques that formulate the suitable functional form for estimation. Bayesian analysis and COCOMO II are their examples.

Remaining sections of the paper are organized as follows: Section II describes the background of the technique i.e. Use Case Point Model including all the formulas needed for calculation. Section III describes the related work in the field of software estimation including all the tools and framework based on Use Case Point model. Section IV shows the conclusion of the paper.

II. BACKGROUND

This section describes Use Case Point (UCP) model in detail including all the necessary steps with its diagrammatical representation.

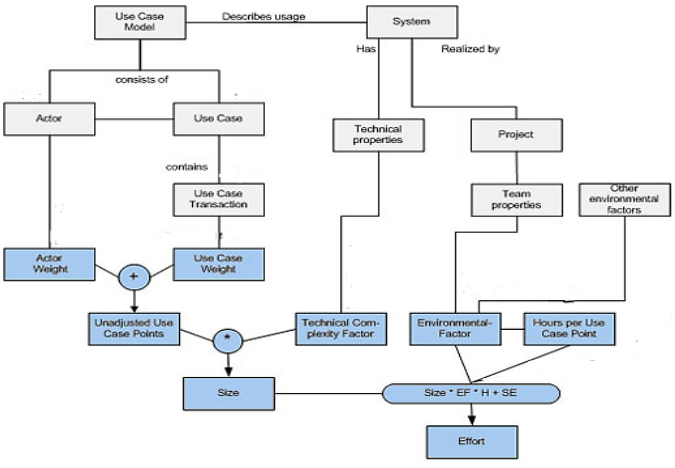

Fig. 1. Use Case Point Model

A. Use Case Point Model

In 1993, Gustav Karner [12] introduced the Use Case Point (UCP) model which is based on use case diagrams for software cost estimation. For effort estimation Use Case Point method follows these steps:

1) Unadjusted actor weight (UAW): All the actors classified into 3 categories i.e. simple, average and complex based on type of interaction with the system. Actor classification rules are:

a) Simple: Interact with system through simple programming interface (API).

b) Average: Interact with the other system using some protocol like HTTP, FTP.

c) Complex: Interact normally through web page or GUI.

2) Unadjusted Use case Weight (UUCW): All the Use Cases classified into 3 categories i.e. simple, average and complex based on number of interaction with the system. Use Case classification rules are:

a) Simple: Less than or equal to 3 transactions or less than 5 classes.

b) Average: Between 4 to 7 transactions or between 5 to 10 classes.

c) Complex: Greater than 7 transactions or more than 10 classes.

3) Unadjusted Use Case Point (UUCP): Unadjusted Use Case Point is obtained by adding Unadjusted Actor Weight and Unadjusted Use Case Weight as shown in "(1)".

$$UUCP = UAW + UUCW \quad (1)$$

4) Adjusted Use Case Point (AUCP): By multiplying UUCP by the Technical Complexity Factors (TCF) and the Environmental Complexity Factors (ECF) AUCP is obtained. Their formulas are given in "(2)" and "(3)".

$$TCF = 0.6 + (0.01 * \text{Total Factor}) \quad (2)$$

$$ECF = 1.4 + (-0.03 * \text{Total Factor}) \quad (3)$$

Where, Total Factor is obtained by adding *(Ti *Wi)* for all respective factors i.e. technical and environment factors where *Wi* is the given weight and *Ti* is a factor that takes values between 0 and 5 where "0" value shows no significance, "1" shows insignificant effect and "2" shows moderate influence, "3" shows intermediate effect, "4" shows significant influence and "5" shows very strong influence. Weights of technical and environment factors are given in Table I and Table II.

TABLE I. TECHNICAL FACTORS [19]

Factor	Description	Weight
T1	Distributed system	2
T2	Performance	1
T3	End-user efficiency	1
T4	Complex internal processing	1
T5	Reusable Code	1
T6	Easy to install	0.5
T7	Easy to use	0.5
T8	Portable	2
T9	Easy to change	1
T10	Concurrent	1
T11	Special security features	1
T12	Direct access to third parties	1
T13	Special user training facilities needed	1

TABLE II. ENVIRONMENTAL FACTORS [19]

Factor	Description	Weight
T1	Familiar with Standard Process	1.5
T2	Application experience	0.5
T3	Object-Oriented Experience	1
T4	Lead analyst capability	0.5
T5	Motivation	1
T6	Stable requirements	2
T7	Part-time workers	-1
T8	Difficult programming language	-1

Finally, the Adjusted Use Case Points (UCP) can be defined as follows:

$$AUCP = UUCP*TCF*ECF \quad (4)$$

5) Effort estimation: By incorporating TCF and ECF, the value of UCP we get the effort. For effort estimation, Karner [12] was given 20 person-hours to develop each UCP. This is expressed in "(5)":

$$Effort=AUCP*20 \quad (5)$$

Where, Effort is measured in person-hours and AUCP is measured in UCP.

III. RELATED WORK

In 90's to estimate effort using Use Case Point, primary requisite is to identify and gather data. So, to fulfill this requirement Shepperd et al. [32], [31] proposed an automated PC based tool known as ANGEL (Analogy) software tool. ANGEL is a flexible tool which is based upon the minimization of Euclidean distance in n dimensional space and responsible for storage, collection & identification of data. Moser et al. [24] developed a new approach, System Meter(SM) method which mainly focused on reuse criteria. For evaluation purpose authors taken 36 projects from the Swiss industry & results showed that SM is better than FPA for effort estimation.

Nowadays organizations are changing with great speed. As the need of accurate estimation increases day by day, researchers gained attention towards introducing the tools and frameworks to meet the requirements of industries. So, many researchers have done work in this direction and successful in doing so like Kusumoto et al. [20], [21] proposed an automatic use case measurement tool known as U-EST (use case based estimation supporting tool). Author analyzed U-EST by a case study & compared the results obtained by U-EST & ones calculated by an expert of use case counting. Then, discuss the validity & limitation of the results. Finally, author concluded that U-EST is more adequate then the existing method.

Rakovic [28] introduced a methodology which assumes Unified Software Development Process (USDP) and the Unified Modeling Language (UML) based Object-Oriented approach. A case study also performed on POST (Point-of-Sale Terminal) software development project. Results of case study showed that early software projects can be simplified and improved by using proposed method or other methods which are Iterative, Use Case driven and Architecture-centric. In this [10], Monika presented a self-implemented tool containing all functionalities at one location and performs comparison among UCP, COCOMO & COCOMO II. This tool is used in collection of requirements in object oriented paradigm. Finally, results showed that UCP is better than COCOMO & COCOMO II in terms of effort and time.

Some metrics are also proposed by some authors for estimating effort which are based on Use Case Point. Braz et al. [8] introduced two metrics namely, USP (Use case Size Point) which taken into consideration internal structure of use case point & FUSP (Fuzzy Use case Size Point) which taken into consideration fuzzy set theory to deal with uncertainity. Both of these proposed metrics used to overcome the limitations of use case point (UCP) method. To show the applicability of proposed metric & its advantages an empirical evaluation is performed. Results demonstrated that USP & FUSP give better estimated value than use case point (UCP) approach.

Grimstad et al. [13] illustrated through a real life example that the lack of proper estimation error technique which is used for analysis purpose may lead to wrong results. So, to overcome this problem authors proposed a framework which improves the analysis of software cost estimation error.

In [9] authors used the use case point method for the effort estimation or avionics systems. In this paper, authors proposed EPCU tool which act as software estimation tool & used it in a case study. At the end author concluded that UCP technique gives satisfactory results i.e. estimated effort is close to the actual result.

Wang et al. [11] proposed extended use case point approach by integrating fuzzy set theory and BBNS (Bayesian Belief Networks) for software cost estimation to address the issue of use case point method. At the end authors compared the UCP& EUCP methods and analyzed the proposed work with the help of two case studies.

Nassif et al. [4] discussed about the existing use case model including its disadvantages also. And to overcome these disadvantages authors proposed an enhanced use case point method for software cost estimation using soft computing techniques like neural network approach and fuzzy logic approach. Final results showed that for some projects use case point for software estimation can be improved by 20%.

Nassif et al. [1] proposed a regression model that is based on the non- linear relationship between software effort and size of software and affected by productivity and complexity of the project. Result analysis is also performed by using MMRE & PRED (.25) which improved the accuracy of estimation by 16.5%. Later, Ali bou nasiif et al. [14] provided an overview of several estimation techniques that are expert judgment, algorithmic model & machine learning and then discussed about the various steps involved in UCP method. They introduced a regression model based on the use case point method for effort estimation and they also presented a Sugeno fuzzy logic approach for the improvement of the proposed regression model. Results showed that software effort can be increased 11% by using UCP regression model. In future, instead of sugeno fuzzy logic approach we may use another fuzzy approach or neural network approach.

Nasiff et al. [3] proposed a regression model using mamdani fuzzy inference system. This model is used for estimating the effort of software in early stages and based on use case point method. After evaluating the results it is concluded that the applying proposed regression model gives overall 10% improvement.

Nassif et al. [6] proposed a Treeboost model which is also known as Stochastic Gradient Boosting (SGB) the purpose of software effort estimation and based on software size, complexity and productivity. Boosting is an approach which is used to enhance the accuracy of a function by applying the function in a series and adding the output of all the individual function. This model is applicable only for those projects having around 2475 UCP. Authors also used MMRE, PRED, MSE, MdMRE for the evaluation purpose. In the results it is proved that Treeboost model is better than both regression model as well as UCP model.

Kamal et al. [18] performed literature survey and discussed about the advantages and limitations of the existing approaches with respect to some attributes like accuracy, ease of use, sensitivity, transparency, adaptability, empirical validation etc. After analysis of the existing approaches like Use Case Points (UCP), Extended Use Case Points (EUCP), UCPm, Adapted Use Case Points (AUCP), Use Case Size Points (USP), Fuzzy Use Case Size Point (FUSP), Simplified Use Case Points (SUCP), Industrial Use Case Points (IUCP) authors proposed a new framework for effort estimation which is use case-based model using fuzzy logic.

Jha et al. [15] presented a new framework for the cost estimation using parameterized for use case point model (P-UCP). In this framework, authors explore the use case model that consist use case point narrative & sixteen new environment factors. In short, we can say that UCP is calculated by including new parameters.

According to Zhen li et al. [34] SOA systems are more complicated so to estimate cost & effort for these systems are challenging task & traditional cost estimation techniques are inadequate to address these systems. So, authors proposed a divide and conquer approach & illustrated it with the help of an application case study. Proposed approach has many merits like structural simplicity, computational efficiency, parallelism & capability of solving complexity.

Robiolo et al. [29], [30] presented an approach for early estimation which is based on use cases. In this paper, entity objects & transactions are the notions for size which is used for estimation. For performance analysis a case study is performed by the author and the results proved that using transactions as a notion of size is beneficial for the improvement of effort estimation.

Jha et al. [16] introduced six new environment factors for making the results of use case point method more precise and accurate. Authors assign weight to all the new six environment factor based on the past experience and historical data. This paper also shows that weight values should be chosen precisely because even a small fluctuation in weights can change the results by 40% and lead to catastrophic results.

Mohagheghi et al. [23] used existing method for incremental software development and evaluate on large scale industrial project from the previous release with modifications. In this paper, authors modified some elements of original method that are management of non – functional requirements & factors of team that may affect effort and actors & use cases complexity assessment. The approach used data from the one release & generated an estimate value for the successive release that was 17% less than the actual effort. For the large software products with incremental development identified factors which may affect their effort.

Nassif et al. [5] presented two approaches namely, log-linear regression and a multilayer perceptron (MLP) neural network model. Former one is based on Use Case Point (UCP) for effort estimation & to calibrate the productivity factor in the regression model fuzzy logic approach is used. Later one is based on team productivity & software size. Author performed comparison of these two approaches on the basis of project size. Results showed that, for the smaller projects MLP model is better and for larger projects Log – linear regression model is better.

Jose et al. [27] for effort estimation in software development projects apply estimation technique to each iterative incremental life cycles of project it gives better result. For this purpose authors used the COCOMO II and Function Point Analysis (FPA) approaches.

Nagar et al. [26] reviewed classical approaches for software effort estimation that are used in industries. But, to meet the latest customer requirements authors introduces a systematic approach which considered all the factors that may affect software project effort adversely and cover all the software development stages.

As existing estimation methods are suitable for traditional sequential software methodology but for agile software development they will result in inaccurate estimation. So,

Ziauddin et al. [35] proposed a model to estimate the effort of agile software projects. For the evaluation purpose 21 projects were estimated and concluded that proposed model provided the better estimation accuracy in terms of PRED (n) and MMRE.

Nassif et al. [2] proposed a feed-forward artificial neural network (ANN) model to estimate software effort based on the use case point method. Software size, project complexity & productivity were taken as the input and model were trained using 168 projects and evaluated using 72 projects against regression as well as UCP method. MMRE & PRED criteria were used for evaluation purpose. At the end, results showed that ANN model outperforms the UCP & regression model by 50% & 8% based on MMRE & PRED for .50,.75,1 except for Pred (0.25) UCP method is better.

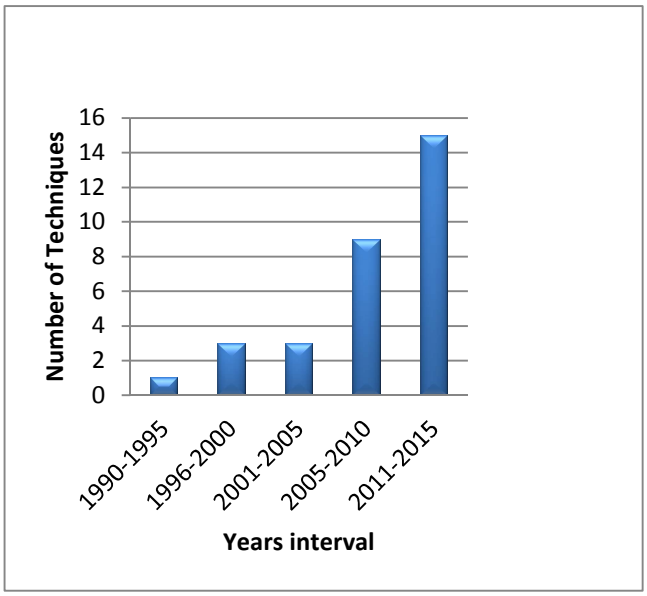

Fig. 2. Techniques evolved over the years

Mohammad et al. [22] proposed a fuzzy model tree for an early effort estimate which is also based on the Use Case Point method. This proposed method compared with the other existing models, Treeboost model and multiple linear regression models. After evaluation by using different criteria results showed that the performance of proposed one is better than the existing one approach for effort estimation.

Wei Lin Du et al. [36] proposed a hybrid intelligent model with combination of neural network model & fuzzy model (neuro fuzzy model) which enhance the accuracy of software cost estimation. Industrial data and published projects were taken as the input for evaluation purpose. MMRE criteria used for evaluation that showed improved estimation with 18% accuracy.

From the above discussion, we concluded that evolution of number of tools and frameworks for software effort estimation increases day by day as software effort estimation is primary requisite software development process. Fig.2. shows the statistical representation of evolution of different tools and techniques over the years. For this representation we have taken into consideration all the tools and frameworks mentioned in this paper and developed from 1990 to 2015 year.

IV. CONCLUSION

This paper has presented an overview of various software effort estimation techniques. In this research paper, we have concluded that UCP is reliable as other existing approaches such as COCOMO, FPA. Use Case Point is a valuable addition to the tools available for effort estimation and various tools are based on UCP model. In future, we can integrate some fuzzy logic and neural network approaches that are not used till now to get better estimate. We can also introduce model that also support projects which are develop in agile environment.

REFERENCES

[1] A. B. Nassif, D. Ho and L. F. Capretz, "Regression model for software effort estimation based on the use case point method," in 2011 International Conference on Computer and SoftwareModeling, Singapore, 2011, pp. 117-121.

[2] A. B. Nassif, L. F. Capretz and D. Ho, " Estimating Software Effort Using an ANN Model Based on Use Case Points", 11th International Conference on Machine Learning and Applications, DOI 10.1109/ICMLA.2012.138, IEEE computer society, 2012.

[3] A. B. Nassif, L. F. Capretz and D. Ho, "A regression model with mamdani fuzzy inference system for early software effort estimation based on use case diagrams", in Third International Conference on Intelligent Computing and Intelligent Systems, Guangzhou, Guangdong, 2011, pp. 615-620.

[4] A. B. Nassif, L. F. Capretz and D. Ho, "Enhancing Use Case Points Estimation Method using Soft Computing Techniques," Journal of Global Research in Computer Science, vol. 1, pp. 12-21, 2010.

[5] A.B. Nassif, D. Ho and L. F. Capretz. "Towards an early software estimation using log-linear regression and a multilayer perceptron model," Journal of Systems and Software, 2012, DOI: 10.1016/j.jss.2012.07.050, Elsevier.

[6] A.B. Nassif, L.F. Capretz, D. Ho and M. Azzeh, "A Treeboost Model for Software Effort Estimation Based on Use Case Points", ACM, In Proceedings of ICMLA (2). 2012, 314-319.

[7] B. Boehm, C. Abts, and S. Chulani, "Software development cost estimation approaches - A survey," Ann. Softw. Eng., vol. 10, 2000, pp. 177-205.

[8] Braz, M. R. and Vergilio, S. R., "Software Effort Estimation Based on Use Cases", Proceedings of the 30th Annual International Computer Software and Applications Conference (COMPSAC '06), 2006, pp. 221228.

[9] Caio Monteiro Barbosa da Silva, Denis Silva Loubach and Adilson Marques da Cunha, "Applying the use case points effort estimation technique to avionics systems", 978-1-4244-2208-1/08/, IEEE, 2008.

[10] Dureja, M., "Performing evaluation of COCOMO vs. UCP", International Journal of Computers and Technology, Vol. 13, No 4, April 30, 2014, ISSN 2277-3061.

[11] F. Wang, X. Yang, X. Zhu and L. Chen, "Extended use case points method for software cost estimation," in International Conference on Computational Intelligence and Software Engineering, 978-1-4244-4507-0/09,2009.

[12] G. Karner, "Resource Estimation for Objectory Projects," Objective Systems, 1993.

[13] Grimstad, S. and Jørgensen, M., "A Framework for the Analysis of Software Cost Estimation Accuracy". ISESE'06, September 21–22, 2006, Rio de Janeiro, Brazil, ACM.

[14] I. Jacobson, M. Christerson, P. Jonsson, and G. Overgaard, "Object-Oriented Software Engineering: A Use Case Driven Approach", Addison Wesley Longman, Inc, 1992.

[15] Jha, P. Malu, R. K., "Cost estimation using parameterized for use case point model (P-UCP)", Int. Journal of Engineering Research and Applications, ISSN: 2248-9622, Vol. 3, Issue 5, Sep-Oct 2013, pp.1280-1283.

[16] Jha, P., Jena, P.P. and Malu, R. K., "Estimating software development effort using UML use case point (UCP) method with a modified set of environmental factors", International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, 2742-2744, ISSN: 0975-9645.

[17] Jyoti G. Borade and Vikas R. Khalkar, "Software Project Effort and Cost Estimation Techniques", International Journal of Advanced Research in Computer Science and Software Engineering, ISSN: 2277 128X ,Volume 3, Issue 8, August 2013.

[18] Kamal, M. W., and Ahmed, M. A., "A proposed framework for use case based effort estimation using fuzzy logic : building upon the outcomes of a systematic literature review", International Journal on New Computer Architectures and Their Applications (IJNCAA) 1(4): 953-976, The Society of Digital Information and Wireless Communications, 2011 (ISSN: 2220-9085)

[19] Koirala, Shivprasad, "How to Prepare Quotation Using Use Case Points", The Code Project, December 2004.

[20] Kusumoto, S., Matukawa F., Inoue K., Hanabusa S. and Maegawa Y., "Estimating Effort by Use Case Points: Method, Tool and Case Study", Proceedings of the 10 International Symposium on Software Metrics (METRICS'04), 2004.

[21] Kusumoto, S., Matukawa F., Inoue K., Hanabusa S. and Maegawa Y., "Effort Estimation Tool Based on Use Case Points Method", Proc. of Software Metrics, 10th International Symposium, Pp 292-299, 2004.

[22] M. Azzeh and A. B. Nassif, "Fuzzy Model Tree For Early Effort Estimation", 12th International Conference on Machine Learning and Applications, DOI 10.1109/ICMLA.2013.115, IEEE computer society, 2013.

[23] Mohagheghi, P., Anda, B., Conradi, R., "Effort Estimation of Use Cases for Incremental Large-Scale Software Development" ICSE 05 May 15-21, Copyright ACM 1-58113-963-2/05/0005.

[24] Moser, S., Henderson-Sellers, B. and Misic, V., "Cost estimation based on business models", The Journal of Systems and Software 49 (1999) 33-42, 1999 Elsevier.

[25] N. M. Awan and A. K. Alvi , "Predicting software test effort in iterative development using a dynamic Bayesian network", Master Thesis, School of Engineering at Blekinge Institute of Technology, Sweden.

[26] Nagar, C., Dixit, C., "Software Efforts and Cost Estimation with a Systematic Approach", Journal of Emerging Trends in Computing and Information Sciences, Volume 2 No.7, July 2011, ISSN 2079-8407.

[27] Pow-Sang, J.A. and Jolay-Vasquez, E., "An Approach of a Technique for Effort Estimation of Iterations in Software Projects", Proceedings of the XIII Asia Pacific Software Engineering Conference (APSEC '06), pages: 367-376, DOI: 10.1109/APSEC.2006.20.

[28] Rakovic, R. M., "Towards a Methodology to Estimate Cost of Object-Oriented Software Development Projects", ComSIS Vol. 1, No. 2, November 2004.

[29] Robiolo, G. and Orosco, R., "An Alternative Method Employing Uses Cases for Early Effort Estimation", Software Engineering Workshop, (SEW), 31st IEEE; 01/2007.

[30] Robiolo, G. and Orosco, R., "Employing use cases to early estimate effort with simpler metrics", Innovations System Software Engineering (2008) 4:31–43, DOI 10.1007/s11334-007-0043-y, 2008 Springer.

[31] Shepperd, M. and Schofield, C., "Estimating Software Project Effort Using Analogies", IEEE Transactions on Software Engineering, VOL. 23, NO. 12, November 1997.

[32] Shepperd, M., Schofield, C. and Kitchenham, B., "Effort Estimation Using Analogy", Proceedings of the 18th international conference on software engineering (ICSE-18), 1996, IEEE.

[33] Wei Lin Du, Luiz Fernando Capretz, A. B. Nassif and Danny Ho, "A Hybrid Intelligent Model for Software Cost Estimation", Journal of Computer Science 9 (11): 1506-1513, 2013, ISSN: 1549-3636, 2013 Science Publications.

[34] Z. Li and J. Keung, "Software cost estimation framework for service - oriented architecture systems using divide and conquer approach", Fifth IEEE International Symposium on Service Oriented System Engineering (SOSE), 2010.

[35] Ziauddin, Tipu, S.K. and Zia, S., "An Effort Estimation Model for Agile Software Development", Advances in Computer Science and its Applications (ACSA), Vol. 2, No. 1, 2012, ISSN 2166-2924, Copyright © World Science Publisher, United States.