I.J. Information Technology and Computer Science, **2015, 01, 67-72**
Published Online December 2014 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2015.01.08

Evaluating the Maintainability of a Software System by using Fuzzy Logic Approach

Ritika Vern
Amity School of Engineering and Technology, Noida, India
Email: ritika.vern@gmail.com

Sanjay Kumar Dubey
Amity School of Engineering and Technology, Noida, India
Email: skdubey1@amity.edu

Abstract—Maintainability is an important quality attribute for almost every quality model. Maintainability of the software is considered as most expensive phase in software development life cycle as it consumes almost major part of the total effort allocated to the software system. Maintainability evaluation is complex due to its imprecise output. This paper proposes a maintainability model by considering its fuzzy aspects. Since fuzzy modeling deals with uncertainty and impreciseness so this paper uses fuzzy methodology and AHP technique to evaluate the maintainability of the model. Object oriented system has taken as case study for maintainability evaluation purpose.

Index Terms— Maintainability, Model, AHP, Software System, Object Oriented.

I. Introduction

Quality of any software depends on the quality factors.

Among all the quality factors, software maintainability is broadly accepted as a highly significant contributor in the economic success of software systems and products. There is a demand for software engineers to understand and urge how various components of a software system is required in order to enhance the reliability of code during maintenance. Maintenance of software is the most expensive and resource requiring phase of the software development process. Statistics from various organizations shows that 40% to 80% of the development expenditure on the average software is spent in the maintenance phase [1, 2]. Maintainability evaluation is an essential component of modern software development life cycle. Maintainability evaluation if done correctly can be useful in software selection and costs estimation among different software, allocation of resource and staff, and so on. This minimizes the future maintenance effort [3].

According to ISO/IEC-9126, maintainability is the capability of the software to be modified [14]. These modifications can be corrective, adaptive or perfective in order for software to command with new requirements and functional specifications.

The ease with which the maintenance of a functional unit can be performed in accordance with prescribed requirements is all about maintainability. There are some major factors which controls maintainability.

The set of maintainability`s sub-characteristics can be classified in a hierarchical tree which consists of the characteristic of maintainability, its sub-characteristics and metrics. Maintainability is on the tree`s highest level while metrics are on the lowest. Maintainability is analyzed by in sub- characteristics which in turn can be evaluated by using metrics.

For evaluating the maintainability of object oriented system, four factors are taken i.e. complexity, coupling, inheritance and message passing. These factors are chosen since they are the design complexity factors and show more impact on the maintainability of object-oriented software system.

These factors are mapped with associated metrics for which Chidamber and Kemerer (CK) software metrics [15] have been used for maintainability evaluation. These metrics are focused on assessing the design of object-oriented system rather than implementation which makes them more suited to object-oriented paradigm as object-oriented design [16].

A case study is taken for the purpose. The complete evaluation of maintainability is done by using AHP and fuzzy methodology. This paper has designed in various sections.

Section 2 focuses on literature survey related with maintainability evaluation.

Section 3 discussed on Fuzzy Maintainability Model and their mentioned factors. Here primary goal is to present ISO/IEC 9126 model`s dimensions and their associate metrics.

Section 4 defines methodology of evaluation. It defines the software and programming language i.e. Java. This software majorly separates the modules of a management system and then membership mapping has been done. At last defuzzification of the comprehensive score has been done. Here AHP technique has played major role. Basically it’s a stepwise evaluation which holds technique, selected software and defuzzification.

II. Literature Survey

To obtain a higher maintainability level in software system, vast and efficient techniques have been implemented since last decades An explanation of random techniques and methodologies that were in always limelight has provided further. K.K Aggarwal [4] proposed a fuzzy model that was inherited from basic fuzzy interference model. He provided a new Rule Base presentation of Fuzzy model which occupied less storage space. He considered software maintainability factors- RSC, DOQ and UOS. The remarkable step was to validate the fuzzy output which completely justified whole results. Melis Dagpinar [5] worked on particular object oriented metrics for predicting the maintainability of software. An empirical study has done on historical data and metrics data. This paper concluded that direct size and coupling metrics are better maintainability predictors. Felix Garcia [6] proposed metric based evaluation and validation of maintainability. This paper focused on software measurement which is a fundamental factor in evaluation of model by its maintainability. Metrics are used to evaluate the software process model maintainability. For validation, a set of representative metrics has been proposed. Harish Mittal [7] introduced software maintainability assessment which is done by considering four major aspects of software i.e average no. of live variables, average life span of variables, average cyclometic complexity and comment ration. An easy approach of fuzzy logic is used with triangular membership functions using complexity and coefficient metrics. Comparison of 9 projects and their results is accompanied with the validation. Yogesh Singh [8] worked on maintainability and applied the same logic as K.K Aggarwal [4] did, but with little changes i.e. he considered one more maintainability factor ACC (Average Cyclometic Complexity). This paper described a proposed fuzzy model which is being validated with defuzzification with Center Of Gravity method. Methodology included MATLAB oriented values for output. Robert Baggen [9] worked on Code analysis and quality for software maintainability assessment. Approach was supported by quality model ISO/IEC 9126. The backend was working on one criterion that when minimum level of software maintainability is reached, the certification body will provide a Product Maintainability Certificate. This procedure was followed by standardized evaluation. Role of central benchmark repository which supports the evaluation results are explained. Hamdi A. AI-Jamimi [10] proposed a model based on object oriented metrics data in Li & Henry`s datasets. In all related experiments datasets were evaluated on basis of training set & validation set. Fuzzy Logic & CK metrics were the major role models. These experiments were conducted based on significant independent variables. Apart from metric based evaluation Yajnaseni Dash [11] worked on evaluating maintainability by artificial neural network. Maintenance effort and Object oriented metrics were considered as dependent variable & independent variable respectively. Methodology included Multilayer Perception neural network model. Results are further compared with each other model. MLP is found to be efficient for estimation evaluation. Xue-Liang Hao [12] emphasized Software Maintainability evaluation by fuzzy integral evaluation approach. Virtual maintenance system considered as real time example. Methodology was based on Software maintainability metric by fuzzy integral. This paper proposed evaluation model of software maintainability and software maintainability index system. Kiranjit Kaur [13] worked on Multivariate linear model which estimated the maintainability of a class diagram in the term of reliability and portability. Metrics were chosen as CK and MOOD that may help software designer for improving the maintainability of a class diagram in design phase.

III. Fuzzy Maintainability Model

Maintenance is a vital part of development with required and updated special techniques. Most often maintainability is performed without requirements or design documents or it may need basic code as well. Therefore these models are made to perform whole process orderly and efficiently In order to deal with real time applications with maintenance below is the proposed Fuzzy Maintainability model. The purpose of this study is to propose a evaluation model based on fuzzy evaluation approach and using the dimensions of ISO/IEC 9126.

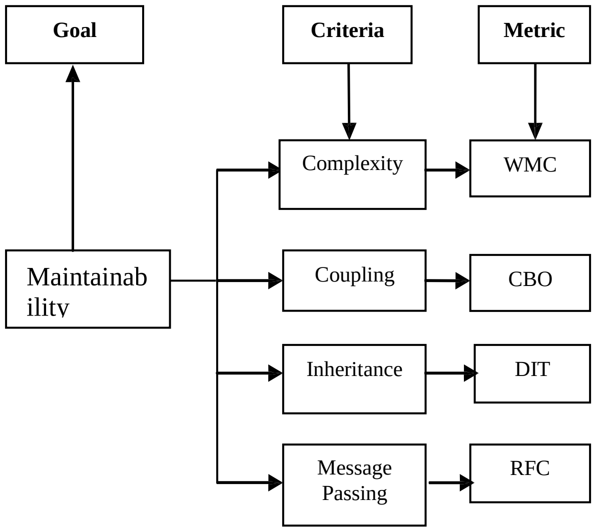

Fig 1. Fuzzy Maintainability Model

This model is showing the metrics that are associated with the sub characteristics (Analyzability, Changeability, Stability and Testability) of Maintainability and will be evaluated further in terms of their metrics that are shown in Fig 1. Scope of studying this Fuzzy Maintainability model is the ease of evaluating Maintainability in Object oriented systems. The further results will be fuzzified. The model accesses modules comprehensively with defuzzified score. Brief outlines of all these factors are shown below [19, 20]:

Complexity defines the difficulty to preserve, modify and comprehend the software. Increased software complexity means that maintenance and enhancement projects will take longer, will cost more, and will result in more errors [21]. It is one of the parts in software metrics that is focused on direct measurement of software attributes, as opposed to indirect software measures such as project milestone status and reported system failures.

Coupling defines the interdependency between components or functions. Coupling is the measure of interconnections among the modules in a software structure. A the degree to which each program module depends on the other and it is required to achieve low coupling in software systems[22]. Coupling is a measure of interdependence of two objects. CBO for a class is measured by counting the number of other classes to which it is coupled. Two classes are coupled if methods of one use methods and/or instance variables of the other. High CBO indicates complex design, decreases modularity, and complicates testing of the class [17].

Inheritance is defined as classes having same methods and operations based on hierarchy. It is a mechanism whereby one object acquires the characteristics from one or more other objects. It is the length of the longest path from a given class to the root class in the inheritance hierarchy and is measured by the number of ancestor classes. So this metric calculates how a class is declared in the inheritance hierarchy. High DIT indicates greater design complexity and more fault-proneness [23].

Message Passing processes or objects can send and receive messages (having zero or more bytes, complex data structures or even segments of code) to other objects or processes. Processes can be synchronized by waiting for messages. It is used for the objects to gather them for a program to work. For objects and systems located on different computers, this is the source for interaction. Message Passing reduces the amount of coding logic and increases maintainability.

In the mentioned proposed model CK metrics are considered. CK metrics are prior when it comes to maintainability evaluation, as their metrics can be used to analyze Coupling, Cohesion, Size very well.

The CK metric suite consists of six design complexity metrics- WMC, CBO, DIT, RFC, NOC and LCOM. The CK metric are validated theoretically in [24, 25]. Various experimental studies also validated CK metrics in [26, 27, 28, 29, 30]. Except for LCOM and NOC, all these metrics can be used as maintainability predictors. LCOM and NOC is uncorrelated with the maintainability of the software [31]. According to Basili et al [32], as NOC increases, Maintainability will decrease. LCOM represents cohesion which defines the closeness of operation in a class & with each other. As cohesion triggers up, less error will come [35]. However they concluded fact is that lower productivity of system comes from high values of LCOM. Hence less Maintainability occurs [33].

IV. Methodology

To evaluate maintainability, project on student management system developed in Java [34, 35] is considered. Tools selected as Analyst4j tool. Analyst4j is an Eclipse platform based tool which is available as a stand-alone Client Application or as an Eclipse IDE plug-in. It features search, metrics, analyzing quality, and report generation for Java programs. Analyst4j tool are in demand for finding out the quality related metrics [36]. It generates various reports like shown in Fig 2. The AHP method is applied for judgment procedure [17, 18].

Table 1. Matrix representation for Eigen Vector

	COMP	COUP	INH	MSGP	The root	Eigen Vector(A)
COMP	1	1	1/3	5	1.136	0.226
COUP	1	1	1	3	1.316	0.262
INH	3	1	1	9	2.279	0.454
MSGP	1/5	1/3	1/9	1	0.293	0.058
Total					5.024	1.000

Where COMP-Complexity, COUP-Coupling, INH-Inheritance and MSGP-Message Passing.

$\lambda max = 4.17$, C.I = 0. 056, C.R =0.062

Therefore,

$$A = [0.226\ 0.262\ 0.454\ 0.058] \quad (i)$$

A. Determining the Fuzzy Member Function for Appraisal Matrix R

The following table 2 describes the fuzzy member function. This comprises metric ranges and divided on the scale of their performance .Eg Complexity varies from 0-20. Its associated metric will judge their category (Very poor, Poor, Medium, Good and Excellent) according to their ranges. Similarly, same goes for rest of metrics.

The taken project has 16 modules i.e. AddSubjectsDialoge.java, EditGradeDialoge.java, EditstudentDialoge.java, EditsubjectDialoge.java, EnrollDialoge.java, GradesDialoge.java, Person.java, RegistryDialoge.java, Semester.java, Semesterlist.java, Student.java, Studentlist.java, StuRec.java, Student.java, Viewelements.java and EasierGridlayout.java.

Table 2. The membership mapping for metric score ranking

	Metric	Very Poor	Poor	Medium	Good	Excellent
COMP(0-20)	WMC	0<= x <5	5<= x <10	10<= x <15	15<= x <20	20<= x
COUP(0-5)	CBO	0<=x <1.25	1.25<=x<2.5	2.5<= x <3.75	3.75<= x <5	5 <= x
INH(0-3)	DIT	0<=x <0.75	0.75<= x <1.5	1.5<= x <2.25	2.25<= x <3	3<= x
MSGP(0-1)	RFC	0<=x <0.25	0.25<= x <0.5	0.5<= x <0.75	0.75<= x <1	1<=x

The module gives values for the Student management system project for WMC, DIT, CBO and RFC as 1.25, 2, 6.5, and 1.3 respectively which is shown in following Fig 3. Similarly other modules are calculated.

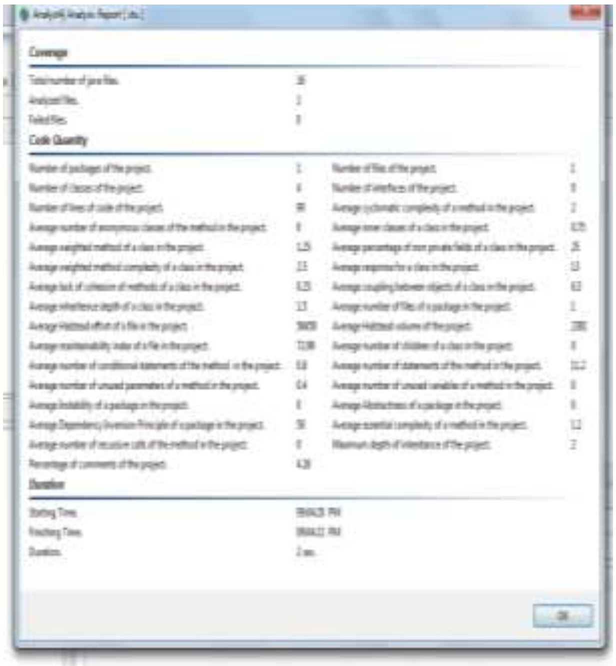

Fig. 2. Snapshot of AddSubjectDialoge.java module

From above snapshot we have values for WMC, CBO and DIT and RFC. Similarly other modules will be calculated in same manner.

B. Determining the Fuzzy Appraisal Matrix

From the Fig 2, i.e. snapshot of a one module, values for each metrics (Complexity, Coupling, Inheritance and Message Passing) can be calculated. Similarly for remaining 15 modules the method will be same. Further these values are compared from the Table 2 which has provided the ranges for four of metrics and then differentiated accordingly for 16 modules. Table 3 is showing the Membership Mapping for the metrics.

Table 3. The Membership Mapping for Task Ranking

COMP	Excellent	Good	Medium	Poor	Very Poor
M1					×
M2			×		
M3				×	
M4				×	
M5			×		
M6					×
M7			×		
M8					×
M9			×		
M10					×
M11	×				
M12					×
M13					×
M14		×			
M15				×	
M16			×		
Total	1	1	5	3	6
Ri	0.0625	0.0625	0.3125	01875	0.375

In similar way **Inheritance, Coupling, Message Passing** is calculated. The final **Appraisal Matrix (R)** is taken by considering all the object oriented factors. The complete matrix is given in Fig.3.The matrix is given as:

$$R = \begin{matrix} COMP \\ COUP \\ INH \\ MSGP \end{matrix} \begin{bmatrix} 0.063 & 0.063 & 0.313 & 0.188 & 0.375 \\ 0.5 & 0.125 & 0 & 0.125 & 0.25 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0.63 & 0.375 & 0 \end{bmatrix}$$

Fig. 3. Matrix representation for Final Appraisal(R)

Where COMP-Complexity, COUP-Coupling, INH-Inheritance, MSGP-Message Passing

Previously Defined, Weight by equation (i) gives

A= [.226 .262 .454 .058]

And so for B= [A] (eqn 1) * [R] (Table 3), the values are

$$\begin{matrix} & b1 & b2 & b3 & b4 & b5 \\ B = [& 0.203 & 0.047 & 0.354 & 0.246 & 0.150] \end{matrix} \qquad \text{(ii)}$$

Where B is the Appraisal Vector . This is the final appraisal vector.

C. Defuzzification

The procedure of converting fuzzy numbers to their approximate crisp numbers is referred to as defuzzication**.** It is defuzzified to a comprehensive score. This paper has defined excellent, good, medium, poor, very poor in appraisal grading as 30, 45, 60, 75, and 90 respectively so appraisal vector B can be defuzzified according to the formula [37, 38]:

$$M_b = \frac{\sum_{i=1}^{5} b_i^2 a_i}{\sum_{i=1}^{5} b_i^2}$$

After defuzzifying, M_b = 60.226

V. Conclusion and Future Scope

Presented paper proposed a model to evaluate the maintainability of object-oriented software system. The inputs for the method were complexity, coupling, inheritance and message passing, which affect the maintainability of the software in different object-oriented software system. These inputs were determined on the basis of survey from different experts project managers, systems developers, researchers and other who are working in this field. A maintainability model is proposed and the maintainability of object oriented system is evaluated based on this model. A fuzzy criteria and AHP technology and fuzzy methodology were used to evaluate the maintainability. The proposed model will be helpful to find the maintainability as a single score for the object oriented software system. Future study will focus on maintainability evaluation of some industrial software system.

REFERENCES

[1] R. Glass, Facts and Fallacies of Software Engineering, Addison Wesley Professional, 2002, pp. 68-73.

[2] R S. Pressman, Software Engineering – A Practitioner's Approach, 7th ed., McGrawhill 2005.

[3] De Lucia A, E. Pompella and S. Stefanucci, "Assessing effort estimation models for corrective Maintenance through empirical studies" Information and Software Technology, vol. 47, no. 1, pp. 3-15, 2005.

[4] K.K Aggarwal, Yogesh Singh and Jitender Kumar Chabbra, "An Integrated Measure of Software Maintainability", in Proceeding of Annual Reliability and Maintainability Symposium, IEEE, 2002.

[5] M. Dagpinar and J. H. Jahnke, "Predicting Maintainability with Object-Oriented Metrics - An Empirical Comparison," presented at 10th Working Conference on Reverse Engineering, British Columbia, Canada, 2003

[6] F. Garcia, F. Ruiz, C. Visaggio, "A Proposal and Empirical Validation of Metrics to Evaluate the Maintainability of Software Process Models" Instrumentation and Measurement Technology Conference, pp. 1093–1097, April 2006.

[7] Harish Mittal & Pradeep Bhatia, "Software maintainability assessment based on fuzzy logic technique" Acm Sigsoft Software Engineering Notes, vol. 34, pp. 1-5, May 2009.

[8] Yogesh Singh, Pradeep Kumar Bhatia and Omprakash Sangwan, "Predicting Software Maintenance using Fuzzy Model", Acm Sigsoft Software Engineering Notes, Vol. 34, pp. 1-6, July 2009.

[9] R. Baggen, K. Schill, and J. Visser, "Standardized code quality benchmarking for improving software maintainability," 4th International Workshop on Software Quality and Maintainability (SQM 2010), March 15, 2010.

[10] H.A. Al-Jamimi and M. Ahmed, "Machine learning-based software quality prediction models: state of the art" Proc. Fourth Int. Conf. on Information Science and Applications, Pattaya, Thailand, 2013.

[11] Y. Dash, S.K. Dubey and A. Rana, "Maintainability Measurement in Object Oriented Paradigm", International Journal of Advanced Research in Computer Science (IJARCS), Vol.3, Issue no.2, pp. 207-213, April 2012.

[12] Xue-Liang Hao, Xiao-Dong Zhu and Li Liu, "Research on Software Maintainability Evaluation Based on Fuzzy Integral", International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering (QR2MSE), 2013.

[13] Kiranjit Kaur and Sami Anand, " A Maintainability Estimation Model and Metrics for Object-Oriented Design (MOOD)", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET),Vol. 2, No 5, May 2013.

[14] Matinlassi Mari and Niemela Eila, "The Impact of Maintainability on Component-based Software Systems", Proceedings of the 29th Conference on EUROMICRO, pp.25, September 2003.

[15] S.R. Chidamber and C.F. Kemerer, "A Metrics Suite for Object-Oriented Design," IEEE Trans. Software Engineering, vol. 20, no. 6, pp. 476-493, June 1994.

[16] S.K. Dubey and A. Rana, "A comprehensive assessment of object oriented software system using metrics approach", International journal of computer science and engineering (IJCSE), pp. 2726-2730, 2010.

[17] S.K. Dubey and A. Rana, "A Fuzzy Approach for Evaluation of Maintainability of Object Oriented Software System", International Journal of Computer Applications, vol.41, issue no. 29, pp. 0975 – 8887, 2012.

[18] C. Jing and L. Xuyan, "Software Maintainability Metrics Based on the Index System and Fuzzy Method", Proceedings of first International Conference on Information Science and Engineering, pp. 5117-5120, 2009.

[19] Amandeep Kaur and Satwinder Singh, "Empirical Analysis of CK & MOOD Metric Suit". Int. Journal of Innovation, Management and Technology, vol. 1, No 5, pp. 447-452, 2010.

[20] D.M. Coleman, D. Ash, B. Lowther and P.W. Oman, "Using Metrics to Evaluate Software System Maintainability," Computer, vol. 27, no. 8, pp. 44-49, August 1994.

[21] BR Sinha, PP Dey, M Amin and H Badkoobehi, "Software complexity measurement using multiple criteria", Journal of Computing Sciences in Colleges, Vol. 28, Issue no. 4, pp. 155-162, April 2013.

[22] R Kumar and D Gupta, "A Heuristics Based Review on CK Metrics", International Journal of Applied Engineering Research, vol.7, No.11, 2012.

[23] S.K. Dubey and A. Rana, "Analytical Evaluation Of Usability of Object Oriented Software System", International Journal Of Computer Applications, vol. 49, Issue no.21, 2012.

[24] G. Poels and G. Dedene, DISTANCE: A Framework for Software Measure Construction, Research Report DTEW9937, Dept. Applied Economics, Katholieke Universiteit Leuven, Belgium, 1999, pp 46.

[25] L C Briand, S. Morasc and V. Basili, " Property-Based Software Engineering Measurement" , IEEE Transactions on Software Engineering, vol. 22, Issue no. 6, pp. 68-86,1996.

[26] B Unger, L Prechelt, and M Philippsen , The impact of inheritance depth on maintenance tasks – Detailed description and evaluation of two experimental replications, Technical Report, Karlsruhe University: Karlsruhe, Germany, 1998.

[27] G Poelsand and G. Dedene, "Evaluating the Effect of Inheritance on the Modifiability of Object-Oriented Business Domain Models",In proceedings 5th European Conference on Software Maintenance and Reengineering (CSMR 2001), Lisbon, Portugal, pp. 20-29,2001.

[28] J. Daly, A Brooks, J Miller, M Roper and M. Wood, "An Empirical Study Evaluating Depth of Inheritance on Maintainability of Object Oriented Software", Empirica Software Engineering, vol. 1, Issue no.2, pp.109-132, 1996.

[29] L. Briand, J. Daly, V. Porter and J. Wust, "A Comprehensive Empirical Validation of Product Measures in Object-Oriented Systems," Technical Report ISERN-98-07, Fraunhofer Inst. for Experimental Software Engineering, 1998.

[30] L.C. Briand, J. Wust, J.W. Daly and D.V. Porter, "Exploring the Relationships between Design Measures and Software Quality in Object-Oriented Systems," The J. Systems and Software, vol. 51, pp. 245-273, 2000

[31] S K Dubey and A Rana, "An Assessment of Maintainability Metrics for Object-Oriented Software System", Acm Sigsoft Sen, vol.36, Issue no. 5, 201

[32] V.R Basili, L. C. Briand and W. L Melo, " A Validation of Object-Oriented Design Metrics as Quality Indicators",IEEE Transactions on Software Engineering, vol. 22, Issue no. 10, pp. 751-761,1996.

[33] S.R Chidamber, D.P. Darcy and C. F. Kemerer, "Managerial Use of Metrics for Object-Oriented Software: An Exploratory Analysis", IEEE Transactions on Software Engineering, vol. 24, Issue no.8, pp. 629-637,1998.
[34] http://1000projects.org/java-projects.html. 2014.01.23
[35] http://www.codeswat.com/cswat/index.php 2014.01.24
[36] S K Bagi ,S. Sharma and S. Bansal, "Analysis-of-software-Metrics-Tools-A-Survey-Approach" International Journal of Scientific & Engineering Research, vol. 4, Issue no. 6,pp. 688, June 2013.
[37] G. Coyle, The analytic hierarchy process (AHP). Practical strategy. Open Access Material, 2004.
[38] T.L. Saaty, "Multi criteria decision making: the Analytic Hierarchy process", RWS publications, 1988. Pittsburgh, PA.

Authors' Profiles

Ritika Vern is pursuing her M.Tech in final year (2014) from the Department of Computer Science and Engineering in Amity School of Engineering and Technology, Amity University Uttar Pradesh, India and completed her B.Tech in Information Technology in 2011 from GGSIPU, Delhi, India. Her research area includes Object oriented Software Engineering and Soft Computing.

Sanjay Kumar Dubey is Ph. D. in Computer Science and Engineering. He is Assistant Professor in Department of Computer Science and Engineering in Amity School of Engineering and Technology, Amity University Uttar Pradesh, India. He is member of IET and ACM digital library. He has a rich academics & research experience in various areas of Computer Science. His research areas include Object Oriented Software Engineering, Soft Computing, HCI, Cloud Computing and Data Mining. He has published more than 55 research papers in indexed National & International Journals including ACM and in Proceedings of the reputed International/ National Conferences (including IEEE Explore). He has authored 2 books also. In addition, he has also served as a Technical Program Committee Member of several reputed conferences.

How to cite this paper: Ritika Vern, Sanjay Kumar Dubey,"Evaluating the Maintainability of a Software System by using Fuzzy Logic Approach", International Journal of Information Technology and Computer Science(IJITCS), vol.7, no.1, pp.67-72, 2015. DOI: 10.5815/ijitcs.2015.01.08