



## **A Roadmap of Software System Maintainability Models**

Monika Saini<sup>1</sup>, Mukti Chauhan<sup>2</sup>

<sup>1</sup>Associate Professor, University Institute of Engineering and Technology  
Panjab University, Chandigarh-160014, INDIA

<sup>2</sup>Research Scholar, University Institute of Engineering and Technology  
Panjab University, Chandigarh-160014, INDIA

---

**Abstract:** Software engineering has become one of the emerging research areas with extensive use of software in the industrial world. Software must be well developed and well maintained to adapt to ever changing requirements of the competitive world. The paper is targeted to present the record of various maintainability techniques or models used so far for software and its applications be it object oriented paradigms or web based applications. This paper reveals all about various parameters including both source code metrics and design metrics used for measuring maintainability of software systems by collecting datasets from different fields including industries, research centres and student projects.

**Keywords:** Software maintainability, maintenance, software metrics, models ,tools, techniques.

---

### **I. Introduction**

Software maintainability has become one of the most important concerns of the software industry. Software maintainability is a key quality attribute of software systems [1]. Since the source code for open source software is openly available so considering both quality and maintainability as important factors open source software are superior to closed source software. Maintenance and Maintainability are two different terms. Software maintenance is defined as “the process of modifying a software system or component after delivery to correct faults, improve performance or other attributes, or adapt to a changed environment where as Software maintainability is defined as “the ease with which a software system or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment [2]. This states that maintenance can be regarded as a process performed during SDLC process of a software but maintainability is one of the important quality attribute[3]. Maintainability factor contributes to 40-70% of the cost of software products. Better maintainability leads to reduced maintenance efforts and reduced cost and time.

### **II. Literature Survey**

Several maintainability models/methodologies were proposed to help the designers in calculating the maintainability of software so as to develop better and improved software systems. Starting from 1970s to 2012 various maintainability predicting models or techniques were developed. Bohem (1978) presented quality model giving maintainability as one of its important attribute. Berns(1984) concluded that maintenance factor depends on the level of difficulty to understand the software program. In 1985 Bowen put forward the equation to find out corrective maintainability. Sneed-Mercy Model(1985), Kafura and Reddy Model(1987) Geoferry and Kemerer Model (1991) estimated maintainability using complexity metrics. Robert Grady(1987) stressed over supportability factor to estimate maintainability. Oman *et.al*(1992) Coleman *et.al* (1994) demonstrated that how software maintainability analysis can be used to guide software related decision making. Li.W *et.al* (1993) used concept of regression to calculate maintainability of software systems. Welker K. *et.al* (1997) concluded that MI should not be interpreted in a vacuum rather it should be used as an indicator to direct human investigation. Muthanna *et al.* (2000), developed a maintainability model using polynomial linear regressions. But this model could be applied only for procedural software and not suitable for object-oriented software. Polo *et al.* (2001) used number of modification requests, mean effort per modification request and type of correction to examine maintainability. M. Dagpinar *et .al*(2003) concluded that size and import direct coupling metrics are significant predictors for measuring maintainability of classes while inheritance, cohesion and indirect/export coupling measures are not. Di Lucca *et.al*(2004) provided WAMM (Web Application Maintainability Model) specific to web applications only. Hayes J.H *et.al* (2005) maintainability model categorized software modules as ‘easy to maintain’ and ‘not easy to maintain’. The model helps the developers to identify the modules those are not easy to maintain, before integrating them. Van Koten(2006) *et.al* BN based model to have better prediction accuracy than regression analysis based model for one out of two datasets i.e UIMS and QUES. Rizvi *et.al* (2010) provided MEMOOD model giving improved

maintainability or understandability of class diagrams and Gautam C *et.al* (2011) provided COMPOUND MEMOOD model which is much better than MEMOOD model giving not only understandability but modifiability, scalability and level of complexity of class diagrams that inturn leads to improved maintainability of software. Ruchika Malhotra *et.al*(2012) estimated maintainability using machine learning algorithms and concluded that Group Method of Data Handling (GMDH) network model is one of the best modeling technique to estimate maintainability of software. Alisara Hincheeranan *et.al* (2012) calculated maintainability considering flexibility and extensibility as two subcharacteristics of maintainability. Dubey *et. al* Model(2012) used Multi Layer Perceptron (MLP) neural network model to predict maintainability using UIMS dataset and found model to be more accurate. Table1 gives summarized view of various maintainability models:

**TABLE 1:Summarized view of Maintainability Models**

| S.NO. | YEAR | MODEL                                                                | MAINTAINABILITY METRICS USED                                                                                                                                                               |
|-------|------|----------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1     | 1977 | McCall's Model                                                       | Simplicity, conciseness, and modularity                                                                                                                                                    |
| 2     | 1978 | Bohem Model                                                          | Understandability ,modifiability ,testability                                                                                                                                              |
| 3     | 1984 | Berns Model                                                          | Maintainability Analysis Tool for use with FORTRAN on aVAX                                                                                                                                 |
| 4     | 1985 | Bowen Model giving Corrective maintainability                        | Average number of days to repair code                                                                                                                                                      |
| 5     | 1985 | Sneed- Mercy Model(Fuzzy Model)                                      | Average number of variables, the Average Cyclomatic Complexity (ACC) and the Comments Ratio (CR)                                                                                           |
| 6     | 1987 | Kafura and Reddy Model                                               | cyclomatic complexity as well as six other software complexity metrics                                                                                                                     |
| 7     | 1987 | Robert Grady Model(At HP)                                            | FURPS+ (Functionality, Usability, Reliability, Performance and Supportability(gives maintainability))                                                                                      |
| 8     | 1991 | Geoferry and kemerer Model                                           | Cyclomatic Complexity Density                                                                                                                                                              |
| 9     | 1992 | Oman -Hagemeister Model                                              | Halstead's Effort (aveE) , McCabe's Cyclomatic ComplexityV(G), LOC(Lines of Code)Average CMT(Lines of Comment),Halstead's Volume(aveV)                                                     |
| 10    | 1993 | Li -Henry model based on coupling between classes                    | DIT(Depth of Inheritance Tree),NOC(Number of Children),CBO(coupling Between Objects)RFC,LCOM,WMC,DAC,NOM,                                                                                  |
| 11    | 1994 | Coleman-Oman model                                                   | Halstead's Volume (aveV) , McCabe's Cyclomatic ComplexityV(G), LOC(Lines of Code)Average CMT(Lines of Comment)                                                                             |
| 12    | 1995 | Welker-Oman model(Improved Oman Model )                              | Halstead's Volume(V),Cyclomatic Complexity V(g'),LOC(Lines of Code)                                                                                                                        |
| 13    | 1995 | Dromey's Quality Model                                               | Selfdescriptiveness, Modifiability, Testability as subcharacteristics of maintainability                                                                                                   |
| 14    | 2000 | Muthanna Model based on Polynomial Linear Regression                 | Several Design Level Metrics                                                                                                                                                               |
| 15    | 2003 | Marcela Genero Model                                                 | Size and Structural Complexity metrics of UML class diagrams with understandability time, modifiability time, modifiability completeness, modifiability correctness                        |
| 16    | 2003 | Huffman Hayes model(Observe-Mine-Adopt (OMA))                        | Maintainability product ,Perceived maintainability                                                                                                                                         |
| 17    | 2004 | Lucca- Fasolino WAMM(Web Application MaintainabilityModel)           | Size metrics, Control coupling metrics ,data coupling metrics, Complexity metrics                                                                                                          |
| 18    | 2004 | Kiewkanya Model                                                      | Metrics Discriminant technique,Weighted Score Level Technique (understandability and Modifiability)                                                                                        |
| 19    | 2005 | Hayes- Zaho Model(MainPredMo)                                        | MI(Maintainability Index),LOC(Lines of Code), TCR(True Comment Ratio), AC(Attribute Complexity),CR(Comment Ratio)                                                                          |
| 20    | 2005 | Aggarwal Model(Artificial Neural Network ANN)                        | Eight object oriented metrics(LCOM,DIT,WMC,NOC,RFC,DAC,MPC,NOM),Maintenance Effort                                                                                                         |
| 21    | 2005 | S.C. Misra Model                                                     | Linear and multiple Regression, Co-relation concept.                                                                                                                                       |
| 22    | 2005 | WANG Li-jin, HU Xin-xin Model                                        | Projection Pursuit Regression model(PPR)                                                                                                                                                   |
| 23    | 2006 | Koten-Gray Model(Bayesian Network Maintainability Prediction Model ) | LOC(Line Of Code), SIZE1,LCOM(Lack of Cohesion of Methods), DAC(Data Abstraction Coupling), DIT(Depth of Inheritance),WMC(Weighted Method per Class),MPC(Message Passing Couple),NOM,SIZE2 |
| 24    | 2007 | Zhou -Leung Model MARS (Multiple Adaptive Regression Splines)        | Compared with other Support vector models , Multivariate linear regression models, Regression tree models and Artificial neural network models and proved to be best .                     |
| 25    | 2008 | Prasanth Ganesh Dalton Model                                         | FRT(Fuzzy Repertory Table) followed by Regression analysis                                                                                                                                 |

|    |      |                                                 |                                                                                                                          |
|----|------|-------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| 26 | 2009 | MO. Elish and KO Elish                          | Treenet model using stochastic gradient boosting.                                                                        |
| 27 | 2010 | Rizvi khan<br>Model(MEMOOD)                     | LOC (Line of Code), MI(), NC (Number of Class),NA (Number of Attributes),NM (Number of Methods), Class Diagram           |
| 28 | 2010 | L Ping                                          | Hidden Markov Model                                                                                                      |
| 29 | 2010 | C Jin, JA Liu                                   | Support vector machine                                                                                                   |
| 30 | 2011 | Gautam Kang Model<br>(COMPOUND MEMOOD<br>MODEL) | LOC (Line Of Code), DLOC (Difference Line of Code),MI (Maintainability Index), CC (Cyclomatic Complexity)                |
| 31 | 2012 | Malhotra- Chug Model                            | WMC, DIT, NOC, RFC,LCOM, MPC, DAC, NOM,SIZE2, SIZE1, CHANGE                                                              |
| 32 | 2012 | Alisara Hincheeranan,<br>Wanchai Rivepiboon     | design size, hierarchies, abstraction, encapsulation, coupling, cohesion etc using Maintainability Estimation Tool (MET) |
| 33 | 2012 | Dubey et. al Model                              | DIT, NOC, MPC, RFC, LCOM, DAC, WMC, NOM, SIZE1 and SIZE2 using Multi Layer Perceptron (MLP) neural network model         |

#### IV. Discussion

Several attempts have been made by different authors to estimate and calculate the maintainability. Various techniques, tools and methodologies were evolved with the passage of time. Following techniques have been used so far for calculating maintainability:

- 1) Maintainability Index equation
- 2) Corrective maintainability equation
- 3) Fuzzy Model
- 4) Machine learning algorithms
  - a) Fuzzy inference systems(FIS)
  - b) Adaptive Neuro Fuzzy Inference systems(ANFIS)
  - c) Artificial Neural Network(ANN)
  - d) Bayesian Networks(Hidden Markov Model)
  - e) Group Method of Data Handling (GMDH)
  - f) Genetic Algorithms (GA)
  - g) Probabilistic Neural Network (PNN)
- 5) OMA (Observe Mine Adopt)paradigm
- 6) UML(using class diagrams)
- 7) Projection Pursuit Regression model(PPR)
- 8) Hierarchical multidimensional models and Polynomial regression models
- 9) Data mining using clustering techniques

Out of these Hierarchical multidimensional models and polynomial regression models used at higher level to estimate maintainability depending upon their ease of use and time taken[10]. Oman's Maintainability Index (MI) using source code metrics has gained much popularity to estimate maintainability[8]. Software with MI < 65 considered to be "low" maintainability between 65-85 is "medium" and MI >85 is "highly" maintainable[10]. Machine learning algorithms are also getting popular. From the literature survey it has been observed that there exists different categories of maintainability metrics: Design metrics, code metrics and Specification metrics (includes mainly complexity metrics). Design metrics include both modularity and structural complexity where as large part of metrics fall in the category of code metrics to measure maintainability. It is most important to have good understanding of software characteristics, metrics and environment in which software is to be analyzed. Commonly used metrics to evaluate and calculate maintainability are [17]:

- 1) McCabe's Cyclomatic complexity metric
- 2) Card and Agresti's System complexity metric
- 3) Halstead's metrics
- 4) Kafura Information Flow metrics
- 5) C- K metrics suit
- 6) D-complexity
- 7) Local Complexity

- 8) Knot count
- 9) Structural complexity metrics(Fan- Out)
- 10) Relative Complexity metric

These metrics can be picked up depending upon respective functional requirements .The main aim of metric selection is to select such metrics which are statistically significant and must be relevant. Very little evidences are available over the maintainability models for web based applications[38]. Studies have been conducted and found that there exists strong relation between Object Oriented software metrics and its maintainability. These software metrics can be calculated using various tools such as:

- 1) Analyst4j
- 2) C&K Java Metrics
- 3) CCCC
- 4) Dependency Finder
- 5) Eclipse Metrics Plugin 1.3.6
- 6) Eclipse Metrics Plugin 3.4
- 7) Semmle
- 8) Understand For Java
- 9) Vizz Analyzer
- 10) JHawk
- 11) MAT (Maintainability Analysis Tool)
- 12) MET (Maintainability Estimation Tool)
- 13) Resource Standard Metrics tool(RSM)
- 14) Crystal Flow Tool

## V. Conclusion and future work

Results are obtained by gathering, studying and analyzing the papers from the year 1970-2012.Special care has been taken to select only the relevant studies irrespective of the open source softwares , closed source softwares and web applications. Only 33 papers were included based on the prediction models presented while other papers were based on their validation and comparison between them. It was observed that little work has been done in maintainability of the web applications and open source softwares. Future work is to work with large versions of open source softwares like Open Office ,Mozilla and Firefox and estimate its maintainability using combination of metrics suggested in past developing new maintainability model. Also using Exploratory factor analysis , Principle component analysis models to estimate maintainability at higher level.

## VI. References

- [1] <http://msdn.microsoft.com/en-us/library/ee658094.aspx>
- [2] IEEE Std. 610.12-1990. *Standard Glossary of Software Engineering Terminology*, IEEE Computer Society Press, Los Alamitos, CA, 1993.
- [3] S.W.A. Rizvi and R.A. Khan, "A Critical Review on Software Maintainability Models," Proc. of the National Conference on Cutting Edge Computer and Electronics Technologies, 14 - 16 Feb. 2009, pp. 144 – 148,Pantnagar, India, 2009
- [4] Boehm, B. W., J. R. Brown, H. Kaspar, M. Lipow, G. J. Macleod and M. J. Merit. 1978. Characteristics of Software QWAT? uality. Amsterdam: North Holland.
- [5] McCall, J.A., Richards, P.K., and Walters, G.F.,(1977) "Factors in Software Quality", RADC TR-77-369, Vols I, II, III, US Rome Air Development Center Reports
- [6]G.M.Berns. Assessing software maintainability.*ACM Communications*, 27(1), 1984.
- [7] Bowen, T. P., Wigle, G. B., Tsai, J. T. 1985. Specification of software quality attributes. Tech. Rep. RADC-TR- 85-37, Rome Air Development Center.
- [8] Sneed, H., Mercy, A. (1985), Automated Software Quality Assurance. IEEE Trans. Software Eng., 11Bi,9: 909-916.
- [9] Grady, Robert, Caswell, Deborah(1987), *Software Metrics: Establishing a Company-wide Program*. Prentic Hall. pp. p. 159.ISBN 0138218447.
- [10]Gill Geoffrey K. and Chris F. Kemerer.(1991). "Cyclomatic Complexity Density and Software Maintenance Productivity,"IEEE Transactions on Software Engineering, Dec, pp.1284-1288.
- [11] P. Oman and J. Hagemester, "Metrics for assessing a software system's maintainability," *Software Maintenance*, 1992, pp. 337 - 344.
- [12] W. Li and S. Henry, "Object-Oriented Metrics that Predict Maintainability", Journal of Systems and Software, vol 23, no.2, 1993, pp.111-122.
- [13]D. Coleman, D. Ash, B. Lowther and P. Oman, "Using Metrics to Evaluate Software System Maintainability", *IEEE Computer*; 27(8), pages 44-49, 1994.
- [14] Welker, K. and Oman, P.W., Software Maintainability Metrics Models in Practice, CrossTalk, Nov./Dec.1995, pp. 19-23 and 32
- [15] Geoff R. Dromey's Model, (Feb 1995) (vol. 21 no. 2),IEEE Transaction On Software Engineering, A Model for Software Product Quality

- [16] Dromey, R.G.: Concerning the Chimera. *IEEE Software* 13 (1), pp. 33–43 (1996).
- [17] S. Muthanna, K. Kontogiannis, K. Ponnambalam and B. Stacey, “A Maintainability Model for Industrial Software Systems Using Design Level Metrics”, In Working Conference on Reverse Engineering (WCRE’00), 2000
- [18] M. Genero, M. Piattini, E. Manso, G. Cantone, “Building UML class diagram maintainability prediction models based on early metrics”, *Proceedings 5th International Workshop on Enterprise Networking and Computing in Healthcare Industry*, , IEEE, 2003, pp. 263-275.
- [19] Hayes, J. Huffman, Mohamed, N., Gao, T. The Observe-Mine-Adopt Model: An agile way to enhance software maintainability. *Journal of Software Maintenance and Evolution: Research and Practice*, Volume 15, Issue 5, Pages 297 – 323, October 2003.
- [20] G. DiLucca, A. Fasolino, P. Tramontana, and C. Visaggio. Towards the definition of a maintainability model for web applications. In *Proceeding of the 8th European Conference on Software Maintenance and Reengineering*, pages 279– 287. IEEE Computer Society Press, 2004.
- [21] Kiewkanya, M., Jindasawat, N., Muenchaisri, P.,(2004) “A Methodology for Constructing Maintainability Model of Object-Oriented Design,” *Proc. 4th International Conference on Quality Software*, 8 - 9 Sept., 2004, pp. 206 - 213. IEEE Computer Society.
- [22] Hayes J.H. and Zaho L (2005), “Maintainability Prediction a Regression Analysis of Measures of Evolving Systems”, *Proc.21st IEEE International Conference on Software Maintenance*, 26-29 Sept.2005, pp.601-604.
- [23] C.V. Korten, A.R. Gray, “An application of Bayesian network for predicting object-oriented software maintainability”, *Information and Software Technology Journal* , vol: 48, no : 1, pp 59-67, Jan2006.
- [24] K.K. Aggarwal, Y. Singh, P. Chandra and M. Puri, “ Measurement of Software Maintainability Using a Fuzzy Model”, *Journal of Computer Sciences*, vol. 1, no.4, pp. 538-542, 2005 ISSN 1549-3636 © 2005 Science Publications.
- [25] K. K. Aggarwal, Y. Singh, A. Kaur and R. Malhotra, “Application of Artificial Neural Network for Predicting Maintainability using Object-Oriented Metrics, *World Academy of Science*, pp. 140-144, 2006.
- [26] Subhas Chandra Misra, “Modeling Design/Coding Factors That Drive Maintainability of Software Systems”, *Software Quality Journal*, 13, pages 297- 320, 2005.
- [27] Y. Zhou and H. Leung, "Predicting object-oriented software maintainability using multivariate adaptive regression splines" , *Journal of Systems and Software*, vol. 80, no. 8, pp. 1349-1361,2007
- [28] Wang Li-Jin Hu Xin-Xin Ning Zheng-Yuan Ke Wen-Hua ,“Predicting Object-Oriented Software Maintainability Using Projection Pursuit Regression.”, *Proceedings of the 2005 International Conference on Software Engineering Research and Practice*, SERP ,vol.2,pp.942-946.
- [29] MO. Elish and KO. Elish , “Application of TreeNet in Predicting Object-Oriented Software Maintainability: A Comparative Study”, *European Conference on Software Maintenance and Reengineering*, pp 1534-5351, March 2009, DOI 10.1109/CSMR.2009.57.
- [30] Rizvi S.W.A. and Khan R.A. (2010) “Maintainability Estimation Model for Object-Oriented Software in Design Phase (MEMOOD)”, *Journal of Computing*, Volume 2, Issue 4, April 2010,
- [31] Malhotra *et.al*, Software Maintainability Prediction using Machine Learning Algorithms.” *Software Engineering : An International Journal (SEIJ)*, Vol. 2, No. 2, SEPTEMBER 2012
- [32] L Ping, “A Quantitative Approach to Software Maintainability Prediction”, *International Forum on Information Technology and Applications*, Vol: 1, No : 1, pp : 105-108, July 2010.
- [33] C Jin , A. L. Jin , “Applications of Support Vector Machine and Unsupervised Learning for Predicting Maintainability using Object-Oriented Metrics”, *Second International Conference on Multi Media and Information Technology* , vol 1 ,no : 1, pp 24-27, April 2010.
- [34] Gautam C, kang S.S (2011), “Comparison and Implementation of Compound MEMOOD MODEL and MEMOOD MODEL”, *International journal of computer science and information technologies*, pp 2394-2398.
- [35]Malhotra *et.al*, ” Software Maintainability Prediction using Machine Learning Algorithms.” *Software Engineering : An International Journal (SEIJ)*, Vol. 2, No. 2, SEPTEMBER 2012
- [36] Alisara Hinceeranan and Wanchai Rivepiboon,” A Maintainability Estimation Model and Tool.” *International Journal of Computer and Communication Engineering*, Vol. 1, No. 2, July 2012.
- [37] Dubey *et.al*.”Maintainability Prediction of Object Oriented Software System by Using Artificial Neural Network Approach.” *International Journal of Soft Computing and Engineering (IJSCE)* ISSN: 2231-2307, Volume-2, Issue-2, May 2012.
- [38] Laxmi Shanker Maurya *et.al*, ” Maintainability assessment of web based application.”, *Journal of Global Research in Computer Science*, Vol 3, No. 7, July 2012.