

# Predicting Maintainability Of Open Source Softwares: An Empirical Approach

Mukti Chauhan  
Research Scholar , ME (IT)  
Panjab University (UIET)  
Chandigarh, India

Monika Sharma  
Assistant Professor (IT)  
Panjab University (UIET)  
Chandigarh, India

## Abstract

*Software engineering is one the emerging field that plays a vital role in the development of any software and its use. Open source softwares are becoming popular with the passage of time and their maintainability is one of the major issue that needs to be considered. This paper is focused to predict the maintainability of two open source softwares including JasperReport and Apache. The aim is to study the impact of software metrics like Lines of code, Cyclomatic Complexity and Halstead volume over maintainability of various versions of Java based open source softwares. Software metrics are calculated using JHawk tool. The result illustrated that these metrics have strong composite impact over the maintainability of open source software due to involvement of human and environmental factors.*

## 1. Introduction

Over the past decade, open source softwares are playing significant role in almost every field including government, industry and academics. Open-source softwares are developed in a public, collaborative manner. Even in businesses as well as educational institutions can get advantage from Open Source software. Open-source software refers to software that is free to use and its source code is fully accessible via the Internet [1]. The success of open-source software is having a growing impact on the software industry, for example Linux and the Apache Web server now have a significant share of their respective software markets. In contrast, closed-source software is developed by a single company; the source code is kept secret. It is normally copyrighted or patented and is legally protected as intellectual property.

With growing demand of open source softwares there is need to maintain them. Maintainability of open source softwares has become one of the important concern in software engineering. Maintainability of the software contributes to quality of the software. Software maintainability is the time consuming and most expensive phase of software

product's life cycle. It contributes to 60-70% cost of the software. Several methodologies are being used in software environment to deal with maintainability of softwares irrespective of closed source softwares or open source softwares. Maintainability and maintenance 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]. A maintainable software product is one that is understandable, testable and easy to modify. There are several metrics including consistency, modularity, simplicity, conciseness and selfdescriptives that can be used to derive maintainability (Gilb 1977; Boehm 1978; Perlis 1981; Arthur 1985). However, efforts are still in service to develop and update the existing methodologies to work over maintainability of open source softwares.

## 2. Historical Perspective

Several maintainability models/methodologies were proposed to help the designers in calculating the maintainability of software so as to develop the better and improved software systems. Starting from 1970s to 2012 various maintainability predicting models or techniques were developed. 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[3],[4],[5]. 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. Software with MI < 65 considered to be "low" maintainability between

65-85 is “medium” and  $MI > 85$  is “highly” maintainable[6]. Li.W *et.al* (1993) used concept of regression to calculate maintainability of software systems[7]. 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[8]. Muthanna *et.al.* (2000), developed a maintainability model using polynomial linear regressions[9]. 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[10]. Hayes J.H *et.al* (2005) [11]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[12]. Rizvi *et.al* (2010) [13]provided MEMOOD model giving improved maintainability or understandability of class diagrams and Gautam C *et.al* (2011)[14] 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 inturn leads to improved maintainability of software. Ruchika Malhotra *et.al*(2012)[15] 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 Hincheeran *et.al* (2012)[16] calculated maintainability considering flexibility and extensibility as two subcharacteristics of maintainability. Dubey *et. al* Model(2012)[17] used Multi Layer Perceptron (MLP) neural network model to predict maintainability using UIMS dataset and found model to be more accurate.

### 3. Objective of the Study

The main objectives of this study are:

1) To study and predict the maintainability of open source softwares .Maintainability of two different open source softwares i.e JasperReport(A reporting

tool)with version 1.0.0,2.0.0,4.0.0,5.0.0 and another open source software Apache(A web server) with versions 1.5.3,1.6.4,1.8.0 will be calculated based on parameter Maintainability Index(MI).  
2) To calculate values of software metrics namely McCabe’s Cyclomatic Complexity (C.C), Halstead’s Volume (Hal.VOI.) and Lines of Code (LOC) for softwares and analyzing their impact on maintainability of the software.

### 4. Dataset and Tool used

Though various tools are available for calculating the software metrics of any software including Analyst4j ,C&K JavaMetrics ,CCCC, DependencyFinder , Eclipse Metrics Plugin , Vizz Analyzer, JHawk ,MAT (Maintainability Analysis Tool), MET (Maintainability Estimation Tool) ,Resource Standard Metrics tool(RSM) and Crystal Flow Tool. In our study ,we are using JHawk version 5 tool for calculating the source code metrics of softwares. Two open source softwares have been used. We collected open source software from <http://sourceforge.net/> and <http://java-source.net>, which are two well-established open source software websites. Two Open Source Softwares (OSS) which we used are JasperReport and Apache. JasperReport is a Reporting Tool and the most popular open source reporting engine. It is entirely written in Java and it is able to use data coming from any kind of data source and produce pixel-perfect documents that can be viewed, printed or exported in a variety of document formats including HTML, PDF, Excel, OpenOffice and Word[18].Different versions of JasperReport that are used includes versions 1.0.0, 2.0.0, 4.0.0 and 5.0.0.Another software Apache is HTTP Server, commonly referred to as Apache is a web server software program notable for playing a key role in the initial growth of the World Wide Web. In 2009, it became the first web server software to surpass the 100 million website milestone[19]. Different versions of Apache that are used Apache 1.5.3,1.6.4,1.8.0.

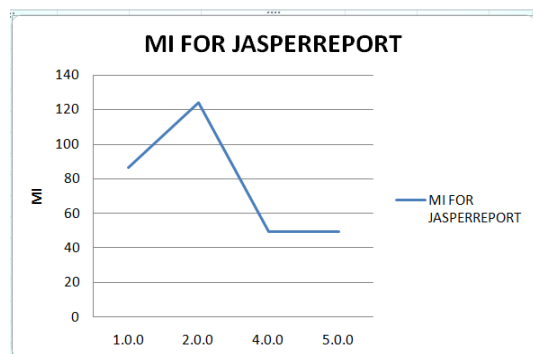
### 5. Experimental Results and Analysis

The value of maintainability for any software can be calculated statistically using maintainability index. This parameter is calculated based on several other commonly available software measures. The maintainability index for two open source softwares was observed with different versions. Firstly, for JasperReport maintainability index for four different versions was observed and result indicated in figure.1that value of MI goes of increasing version 1.0.0 to version 2.0.0 and after 2.0.0 value of MI is

decreasing giving highest value for version 2.0.0 and minimum for version 4.0.0.

**Table 1 . Results for JasperReport**

Versions	Maintainability Index	Average C.C	Hal. VOI	LOC
1.0.0	86.8	0.21	9327	408
2.0.0	124.1	0.08	18326.5	829
4.0.0	49.6	0.74	28855.6	1124
5.0.0	49.65	0.74	29077	1086

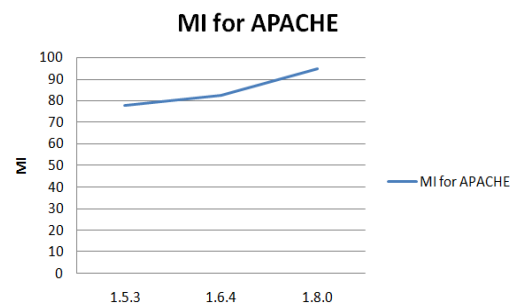


**Figure 1 . Graph giving Maintainability Index of various versions of JasperReport**

Similarly, for apache the we observed the value of MI goes on increasing from version 1.5.3 onwards giving highest value for version 1.8.0. The version with large value of MI is more maintainable so in case of JasperReport version 2.0.0 is more maintainable and less efforts are required for its maintenance. But in case of Apache its maintenance is increasing with release of new versions over a period of time giving highest MI for version 1.8.0. And various software metrics including Average Cyclomatic complexity, Halstead's Volume and Lines of Code play an important role in finding the maintainability of any software. We can observe in both softwares that Average Cyclomatic complexity decreases with increase in value of MI but nothing can be said about halstead's volume and lines of code which indicates that all the three metrics including Average cyclomatic complexity, Halstead Volume and Lines of code have composite effect over the Maintainability Index of the software due to involvement of both human factors and environmental factors.

**Table 2. Results for Apache**

Versions	Maintainability Index	Average C.C	Hal. VOI	LOC
1.5.3	77.85	2	139.47	10
1.6.4	82.65	1	1075.7	45
1.8.0	94.61	0.57	1934.95	67



**Figure 2 . Graph giving Maintainability Index of various versions of Apache**

## 6. Conclusion and Future Work

The work not only analyzed the influences of metrics, but also reported their ability to predict how maintainable a system is, when these metrics are used together. Results have shown these metrics have strong composite effect over the maintainability of open source software. Since Maintainability Index is a composite number and from results it is not immediately clear that how to increase the value of MI when it is low. As in case of JasperReport we observed that MI value increases from version 1.0.0 to version 2.0.0 with decrease in Av.C.C and increase in Vol. and Lines of code. But from version 2.0.0 to version 4.0.0 value of MI is decreasing with increase in value of Av.C.C, Hal.VOI. and Lines of code. So, it cannot be specified that which metric value is effecting MI. And similar is the case for Apache. Hence, we conclude that all the three metrics including Average Cyclomatic Complexity, Halstead Volume and Lines of code have composite effect over the Maintainability Index of the software due to involvement of both human factors and environmental factors.

As different versions of open source softwares, JasperReport and Apache have been analyzed. Few versions of this software have been taken and various metrics have been calculated. But for better results more and more open source softwares with large versions should be taken. The bigger the number of versions is, better will be the results. Future work will

try to focus on the relevancy and extent of factors including both human and environmental factors that affect maintainability of open source softwares.

## 7. References

- [1] [http://www.linfo.org/open\\_source.html](http://www.linfo.org/open_source.html)
- [2] IEEE Std. 610.12-1990. *Standard Glossary of Software Engineering Terminology*, IEEE Computer Society Press, Los Alamitos, CA, 1993.
- [3] Sneed, H., Mercy, A. (1985), Automated Software Quality Assurance. *IEEE Trans. Software Eng.*, 11Bi,9: 909-916.
- [4] Grady, Robert, Caswell, Deborah(1987), *Software Metrics: Establishing a Company-wide Program*. Prentic Hall. pp. p. 159.ISBN 0138218447.
- [5] Gill Geoffrey K. and Chris F. Kemerer.(1991). "Cyclomatic Complexity Density and Software Maintenance Productivity," *IEEE Transactions on Software Engineering*, Dec, pp.1284-1288.
- [6] D. Coleman, D. Ash, B. Lowther and P. Oman, "Using Metrics to Evaluate Software System Maintainability", *IEEE Computer*; 27(8), pages 44-49, 1994.
- [7] W. Li and S. Henry, "Object-Oriented Metrics that Predict Maintainability", *Journal of Systems and Software*, vol 23, no.2, 1993, pp.111-122.
- [8] Welker, K. and Oman, P.W., *Software Maintainability Metrics Models in Practice*, CrossTalk, Nov./Dec.1995, pp. 19-23 and 32
- [9] 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
- [10] 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.
- [11] 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.
- [12] C.V. Koten, 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.
- [13] 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
- [14] 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
- [15] Malhotra *et.al.* " Software Maintainability Prediction using Machine Learning Algorithms." *Software Engineering : An International Journal (SEIJ)*, Vol. 2, No. 2, SEPTEMBER 2012
- [16] Alisara Hincheeran and Wanchai Rivepiboon," A Maintainability Estimation Model and Tool." *International Journal of Computer and Communication Engineering*, Vol. 1, No. 2, July 2012.
- [17] 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.
- [18] <http://community.jaspersoft.com/project/jasperreports-library>.
- [19] [http://en.wikipedia.org/wiki/Apache\\_HTTP\\_Server](http://en.wikipedia.org/wiki/Apache_HTTP_Server).