Commun.Fac.Sci.Univ.Ank.Series A2-A3
Volume 61, Number 1, Pages 57-67 (2019)
DOI: 10.33769/aupse.493048
ISSN 1303-6009 E-ISSN 2618-6462
http://communications.science.ankara.edu.tr/index.php?series=A2A3

QUALITY OF SOFTWARE PROJECTS – A CASE STUDY

Alper KIRAL and Tülin ERÇELEBİ AYYILDIZ

ABSTRACT. In software projects, many points that are overlooked such as time constraints and human factors are causing great problems in the future. By measuring the quality of software projects, problems that may arise in important parameters such as maintenance-repair, functionality and reliability can be eliminated. In this study, metrics that can be used for measuring maintainability quality attribute within the scope of ISO 9126 Quality Standard are examined. In order to perform the study, 40 open source object-oriented software was selected and code complexity analysis was performed. Values of metric sets such as Chidamber and Kemerer (CK), Lorenz and Kidd (LK) and McCabe's complex Suite were determined by the Understand Code Analysis tool. It was determined whether the obtained values exceeded the threshold values indicated in the literature. Frequencies of metrics passing threshold values were determined for 40 open source object-oriented software projects, and the consistency among the metrics was evaluated using WEKA Machine Learning Software and EXCEL Data Analysis Tool. When the results were evaluated, it was observed that in addition to CK metrics such as WMC, CBO, and RFC, which measure the maintainability quality attribute, NOC (CK), NIM (LK), and the ratio of comment/code metrics have been observed to yield significant measurement results

1. INTRODUCTION

As the technology sector becomes a big part of daily life, the software used is constantly expanding as code and manpower. The growth in the software project leads to a significant increase in maintenance costs, project costs and software development time. If these and similar factors cannot be correctly predicted and carried out from the beginning, it is inevitable that problems that cannot be corrected afterwards are encountered. Economic loss of software projects that are rejected by customers, unavailable to use efficiently, canceled due to increased costs, require

Received by the editors: December 06, 2018; Accepted: May 14, 2019.
Key word and phrases: software quality metrics, software quality measurement, maintainability, CK, JK, McCabe, ISO 9126, Understand, WEKA.

high maintenance and repair costs may be far higher than those predicted. As an example,

according to 2002 data, the annual loss of failed software projects to the American economy is around $ 59 billion [1]. In addition, according to Tricentis 2017 research, the loss of failing software in the global economy is around $ 1.7 trillion [2]. By measuring the software quality; early decision can be made for factors that need to be calculated and implemented at early stages such as the ratio of customer needs met by software, the clarity of the software for the developers, the structural quality of the software and the cost and price balance of the software.

Since it is not possible to perform individual code analysis in large projects, there are tools and add-ons that can perform these analyzes in a short time. In this study, 40 open source object-oriented software projects written in Java programming language belonging to Space and Aviation domain were examined with static code analysis tool called Understand. In the scope of ISO 9126 quality standard, the object-oriented metric values which are recommended in the literature are calculated for maintainability quality characteristic and the metric probabilities which are possible to be used in addition to the literature are investigated with the help of WEKA machine learning software.

This study is based on our previous study which investigates the maintainability perspective of software quality metrics [3].

The rest of paper is organized as follows: Section II presents an overview of the related work. In Section III, the metric values analysis and metric threshold exceeding frequencies of 40 open source object-oriented software projects are evaluated. Section IV discusses additional metrics that can be used under the Maintainability quality characteristic. Section V concludes with research results..

2. RELATED WORKS

A. ISO 9126 QUALITY STANDARD

For software quality measurement, various metric clusters are presented by people working in this field. With the help of these metrics, quality requirements are measured. Quality requirements for the ISO 9126 quality standard are shown in Table I [4].

TABLE I. ISO 9126 QUALITY STANDARET AND CHARACTERISTICS[4]

Characteristics	Sub-characteristics
Functionality	Suitability, accuracy, interoperability, security
Reliability	Maturity, fault tolerance, recoverability
Usability	Understandability, operability, attractiveness etc.
Efficiency	Time behavior, resource utilization
Maintainability	Analyzability, changeability, stability, testability
Portability	Adaptability, installaebility, replaceability e.g.

B. SUGGESTED METRICS FOR MAINTAINABILITY QUALITY CHARACTERISTICS MEASUREMENT

One of the metric sets proposed in the literature due to the sub-characteristics of the maintainability attribute includes all Chidamber and Kemerer metrics and additionally the TCC, LCC metrics. [5]. Characteristics, sub-characteristics and metrics are shown in Fig. 1.

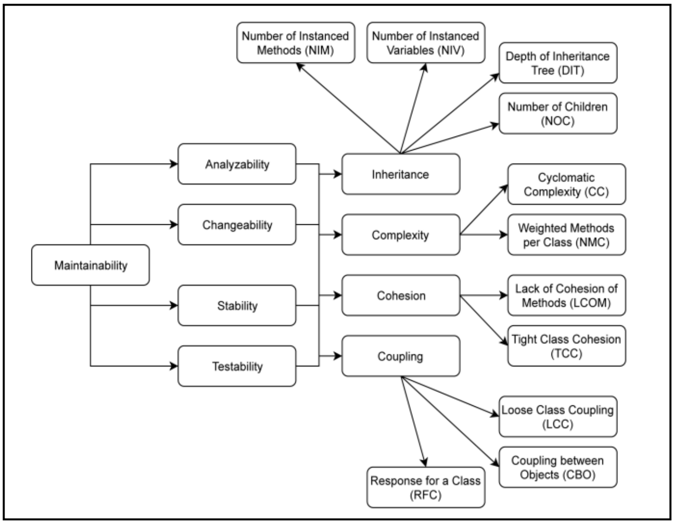

Fig. 1. Maintainability Characteristic and object-oriented metrics

Another suggestion includes WMC, CBO metrics from the CK metric set for the maintainability characteristics measurement, and Table II shows the CK metrics that are suitable for the design phase quality attributes [6].

TABLE II. RELATIONSHIP BETWEEN DESIGN PHASE QUALITY ATTRIBUTES AND CK METRICS

Quality Attribute	**CK Metrics**
Maintainability	WMC, CBO
Reusability	WMC, CBO, DIT, NOC
Testability	RFC, CBO, NOC
Understandability	RFC, CBO, DIT
Development Effort	WMC, LCOM

In this study, by considering WMC, CBO metrics and the importance of machine learning and other metrics, ISO 9126 Standard Maintainability Characteristics will be investigated in the light of Table II and Fig 1. As a result, it is expected that metrics such as DIT, NOC, RFC for reusability and testability attributes will be some of the recommended metrics for measuring the maintainability attribute.

3. PROJECT MEASUREMENTS AND THRESHOLD EVALUATION

Within the scope of the study, 40 open source object-oriented software project written in Java was downloaded from GitHUB and NASA Open Source Software Library. Medium scale projects were chosen for the study, the related projects and total KLOCs are shown in Table III.

TABLE III. PROJECTS AND LINE OF CODES

#	**KLOC**	#	**KLOC**	#	**KLOC**	#	**KLOC**	#	**KLOC**
1	13K	9	52K	17	11K	25	114K	33	25K
2	21K	10	29K	18	44K	26	199K	34	78K
3	11K	11	17K	19	10K	27	124K	35	166K
4	36K	12	49K	20	75K	28	50K	36	20K
5	22K	13	49K	21	15K	29	119K	37	27K
6	54K	14	19K	22	15K	30	68K	38	42K
7	12K	15	14K	23	42K	31	115K	39	10K
8	16K	16	18K	24	63K	32	134K	40	32K

A. DETERMINING THRESHOLD VALUES

Various threshold values were calculated for metrics and used in software quality measurement. Some recommended threshold values for CK metrics are shown in Table IV.

TABLE IV. THRESHOLDS FOR CK METRIC SUITE

	CK Metrics					
Related Works	**LCOM**	**DIT**	**CBO**	**NOC**	**RFC**	**WMC**
[7]	3	6	9	3	6	30
[8]	1	6	8	6	35	15
[9]	Low	4	8	6	35	11
[10]	20	2	14	2	44	20
[11]	Low	4	94	5	10	108
[12]	Low (%85.6)	6	10	14	62	25

Since Reference [12] derives CK metric thresholds over projects in the same domain, this study was continued by using these metric thresholds.

From the McCabe's Complexity Suite metric set, the Cyclomatic Complexity (CC) metric threshold is set to 10, and the Essential Complexity (EC) metric threshold is set to 4 [13-16]. Also recommended Ratio Code/Comment (Ratio C/C) metric threshold value is set to 0.16 [16].

NIV and NIM metrics, which are class metrics from the Lorenz and Kidd metric set, have been selected [14]. A value of 0.8 for the NIM metric threshold and 9 for the NIV metric threshold is suggested [15].

B. CLASS FREQUENCIES EXCEEDING THE THRESHOLD VALUES IN PROJECTS

For the threshold values specified in the previous section, the ratio of the class numbers exceeding the threshold values to the total class numbers in the projects is shown in Table V. Reference [12] showed that 10% of the frequencies did not exceed much for this study domain. Therefore, projects with CBO and WMC frequency less than 10% for WEKA machine learning operations are classified as 0 (successful) and those above as 1 (unsuccessful).

TABLE V. FREQUENCIES FOR CLASSES EXCEED THRESHOLD VALUES

Project #	CBO	NOC	RFC	DIT	LCOM	WMC	CC	EC	Ratio C/C	NIM	NIV	Classification
1	0.05	0.00	0.00	0.00	0.06	0.05	0.04	0.01	0.86	0.15	0.21	1
2	0.02	0.00	0.00	0.00	0.06	0.08	0.02	0.02	0.32	0.14	0.13	1
3	0.08	0.04	0.32	0.00	0.14	0.13	0.02	0.02	0.83	0.01	0.11	0
4	0.06	0.00	0.04	0.00	0.13	0.15	0.01	0.02	0.81	0.13	0.14	0
5	0.03	0.00	0.05	0.00	0.13	0.08	0.02	0.02	0.59	0.05	0.10	1
6	0.04	0.00	0.02	0.00	0.06	0.06	0.00	0.02	0.76	0.16	0.05	1
7	0.02	0.00	0.00	0.00	0.02	0.03	0.02	0.02	0.66	0.31	0.06	1
8	0.05	0.00	0.00	0.00	0.11	0.05	0.01	0.03	0.84	0.19	0.13	1
9	0.12	0.00	0.01	0.00	0.06	0.08	0.00	0.02	0.81	0.13	0.05	0
10	0.03	0.00	0.00	0.00	0.07	0.08	0.01	0.01	0.21	0.13	0.07	1
11	0.02	0.00	0.00	0.00	0.07	0.02	0.03	0.02	0.88	0.24	0.12	1
12	0.09	0.00	0.00	0.00	0.04	0.07	0.04	0.07	0.54	0.14	0.10	1
13	0.02	0.00	0.01	0.00	0.09	0.06	0.01	0.03	0.69	0.04	0.10	1
14	0.01	0.00	0.01	0.00	0.08	0.10	0.02	0.02	0.71	0.08	0.12	0
15	0.01	0.00	0.00	0.00	0.08	0.05	0.02	0.03	0.88	0.30	0.05	1
16	0.05	0.00	0.00	0.00	0.07	0.05	0.01	0.02	0.94	0.14	0.07	1
17	0.01	0.00	0.00	0.00	0.09	0.07	0.04	0.04	0.77	0.23	0.20	1
18	0.06	0.01	0.00	0.00	0.05	0.05	0.01	0.02	0.84	0.13	0.05	1
19	0.03	0.00	0.00	0.00	0.09	0.06	0.02	0.03	0.87	0.20	0.09	1
20	0.12	0.01	0.01	0.00	0.11	0.07	0.04	0.04	0.09	0.04	0.15	0
21	0.01	0.00	0.00	0.00	0.12	0.08	0.01	0.01	0.45	0.38	0.10	1
22	0.10	0.00	0.15	0.00	0.07	0.10	0.05	0.01	0.37	0.05	0.18	0
23	0.08	0.01	0.01	0.00	0.07	0.08	0.03	0.05	0.48	0.06	0.18	1
24	0.04	0.00	0.01	0.00	0.08	0.10	0.04	0.06	0.31	0.12	0.15	1
25	0.12	0.00	0.01	0.00	0.04	0.04	0.00	0.02	0.71	0.10	0.04	0
26	0.09	0.01	0.25	0.00	0.06	0.07	0.01	0.04	0.79	0.08	0.05	1
27	0.11	0.01	0.24	0.01	0.09	0.07	0.02	0.05	0.79	0.21	0.04	0
28	0.08	0.00	0.03	0.00	0.12	0.06	0.01	0.02	0.62	0.07	0.07	1
29	0.10	0.00	0.03	0.00	0.14	0.11	0.06	0.06	0.55	0.12	0.12	0
30	0.03	0.00	0.01	0.00	0.10	0.09	0.05	0.05	0.70	0.18	0.23	1
31	0.03	0.01	0.00	0.00	0.08	0.05	0.03	0.03	0.78	0.14	0.08	1
32	0.02	0.00	0.01	0.00	0.11	0.08	0.02	0.02	0.76	0.16	0.15	1
33	0.03	0.00	0.01	0.00	0.07	0.07	0.01	0.02	0.88	0.10	0.10	1
34	0.02	0.00	0.00	0.00	0.08	0.06	0.03	0.01	0.94	0.11	0.14	1
35	0.20	0.01	0.01	0.00	0.10	0.06	0.01	0.01	0.33	0.07	0.12	0
36	0.15	0.00	0.01	0.00	0.07	0.06	0.01	0.01	0.41	0.04	0.08	0
37	0.02	0.00	0.02	0.00	0.09	0.10	0.02	0.02	0.59	0.16	0.18	0
38	0.00	0.00	0.00	0.00	0.23	0.02	0.00	0.00	0.96	0.02	0.01	1
39	0.07	0.01	0.09	0.00	0.14	0.11	0.03	0.06	0.30	0.10	0.22	0
40	0.01	0.00	0.00	0.00	0.02	0.01	0.00	0.01	0.10	0.02	0.26	1

The ANOVA table of multi regression test which belongs to Table V is shown in Table VI. Classifications were used as criterion variables and metric frequency values were used as predictors for the regression test. Sig. F value shows that classification with WMC and CBO metrics is significant for this research.

TABLE VI. ANOVA TABLE

ANOVA					
	df	**SS**	**MS**	**F**	**Sig. F**
Regression	11	5.81	0.53	4.98	0.00029
Residual	28	2.97	0.11		
Total	39	8.78			

4. ESTIMATING NEW MAINTAINABILITY METRICS

40 open source object-oriented software projects are classified according to the amount in which they exceed WMC and CBO metrics as in Table V. These classifiers are included in WEKA machine learning software.

A. ATTRIBUTE SELECTION PHASE

By using classified projects, the correlation and significance between the metrics were investigated by correlation attribute selection evaluation option in WEKA.

TABLE VII. SELECTED METRICS BY WEKA CORRELATION RANKING RESULT

Attribute Evaluator (supervised, Class (nominal): 12 maintainability): Correlation Ranking Filter	
Ranked attributes:	
Score:	**Metric:**
0.637	CBO
0.503	WMC
0.37	RFC
0.306	NOC
0.303	NIM
0.237	Ratio C/C
Selected attributes	1, 6, 3, 2, 10, 9 : 6

6 metrics are chosen to be used in the next phase by ranker selection method as a result of machine learning process. Table VII shows these results. It is already

foreseen that the first two orders should be CBO and WMC. Remaining four recommendations will be examined in following sections.

B. TRAINING AND TESTING PHASE

In the WEKA Classification section, data belonging to the previous WMC and CBO based classification were used as training data and machine learning was performed. By using CBO and WMC metrics a Naïve Bayes classification was made in WEKA and the results are shown in Table VIII. Projects are correctly classified by 80% accuracy.

TABLE VIII. WEKA NAIVE BAYES CLASSIFICATION RESULTS FOR WMC AND CBO METRICS

Correctly Classified Instances	32	80%
Incorrectly Classified Instances	8	20%
Kappa Statistics	0.5616	
Mean Absolute Error	0.2053	
Root Mean Squared Error	0.3967	
Relative Absolute Error	46.1509%	
Root Relative Squared Error	84.1191%	
Total Number of Instances	40	

Detailed Accuracy By Class

TP Rate	**FP Rate**	**Precision**	**Recall**	**F-Measure**	**MCC**	**ROC Area**	**PRC Area**	**Class**
0.769	0.185	0.667	0.769	0.714	0.565	0.865	0.709	0
0.815	0.231	0.880	0.815	0.846	0.565	0.863	0.935	1
0.800	0.216	0.811	0.800	0.803	0.565	0.864	0.862	**Weighted Average**

Confusion Matrix

a	**b**	← Classified As
10	**3**	**a=0**
5	**22**	**b=1**

Afterwards, RFC, NOC, NIM, Ratio C / C metrics were added to CBO and WMC and a new classification was made by Naïve Bayes method too. Results for this classification are shown in Table IX. Projects are correctly classified by 90% accuracy by WEKA. In terms of maintainability, two projects that were previously marked as successful were considered unsuccessful and two projects that were previously unsuccessful were classified as successful.

TABLE IX. WEKA NAIVE BAYES CLASSIFICATION RESULTS FOR WMC, CBO, RFC, NOC, NIM AND RATIO C/C METRICS

Correctly Classified Instances			36		90%			
Incorrectly Classified Instances			4		10%			
Kappa Statistics			0.7721					
Mean Absolute Error			0.1352					
Root Mean Squared Error			0.288					
Relative Absolute Error			30.6223%					
Root Relative Squared Error			61.4892%					
Total Number of Instances			40					
Detailed Accuracy By Class								
TP Rate	**FP Rate**	**Precision**	**Recall**	**F-Measure**	**MCC**	**ROC Area**	**PRC Area**	**Class**
0.846	0.074	0.846	0.846	0.846	0.772	0.943	0.872	0
0.926	0.154	0.926	0.926	0.926	0.772	0.943	0.975	1
0.900	0.128	0.900	0.900	0.900	0.772	0.943	0.942	**Weighted Average**
Confusion Matrix								
a	**b**	← Classified As						
11	**2**	**a=0**						
2	**25**	**b=1**						

5. CONCLUSIONS

The key contribution of this study is the investigation of the software metrics that are required to measure maintainability characteristic. After researching scientific literature for the recommended metrics to measure the maintainability characteristic, 40 object-oriented open source projects examined with the aid of machine learning tools.

It is seen that WEKA machine learning software has the correct classification with 90% accuracy with new six metrics which is better than the classification with WMC and CBO metrics that have 80% accuracy. Having two projects that are successful and misclassified is considered an acceptable result as false positives. Having small number of true negatives does not hinder our judgement about the project. If general accuracy is high, common procedure of reviewing the negatives would only result in diminishing risks rather than the waste of time.

On the other hand, false positives might affect the projects at hand by misguiding the review process thus, increasing risk. This misclassification should remain minimal with increasing number of projects.

As a result, with 90% accuracy score of classification, it is observed that the usage of CBO, WMC, RFC, NOC, NIM and Ratio C/C metrics in the measurement of maintainability characteristic of ISO 9126 Quality Standard gives consistent results. As a future work, we are planning to increase the number of projects and a more comprehensive machine learning study can be carried out and it can be discussed whether or not to add or remove existing metrics.

References

[1] National Institue of Standards and Technology, 'The Economic impacts of inadequate infrastructure for software testing', 2002. [Online]. Available: https://www.nist.gov/sites/default/files/documents/director/planning/report02-3.pdf. [Accessed: 10- Oct- 2018].

[2] Tricentis.com, 'software fail watch 5th edition', 2017. [Online]. Available: https://www.tricentis.com/wp-content/uploads/2018/01/20180119_Software-Fails-Watch_Small_Web.pdf. [Accessed: 10- Oct- 2018].

[3] Kıral, A., Ayyıldız, T.E., Software Metrics Required to Measure Maintainability, *III. International Conference on Theoretical and Applied Computer Science and Engineering* (ICTACSE 2018), Ankara, TURKEY, November 23-24, 2018.

[4] Jung H., Kim S., and Chung C., "Measuring software product quality: *a survey of ISO/IEC* 9126", September/October 2004, IEEE Software, 88-92.

[5] Y. Fitrisia and B. Hendradjaya, "Implementation of ISO 9126-1 quality model for asset inventory information system by utilizing object oriented metrics," 2014 *International Conference on Electrical Engineering and Computer Science* (ICEECS), Kuta, (2014), 229-234.

[6] U. L. Kulkarni, Y. R. Kalshetty and V. G. Arde, "Validation of CK metrics for object oriented design measurement," *3rd International Conference on Emerging Trends in Engineering and Technology*, Goa, (2010), 646-651.

[7] Antony P., "Predicting reliability of software using thresholds of CK metrics", International Journal of Advanced Networking, (2013) 1778–1785.

[8] Yuming Zhou and Hareton Leung, "Empirical analysis of object-oriented design metrics for predicting high and low severity faults," *IEEE Transactions on Software Engineering,* vol. 32, no. 10, , Oct. 2006, 771-789.

[9] Mago J., Kaur P., "Analysis of quality of the design of the object oriented software using fuzzy logic." *International Conference on Recent Advances and Future Trends in Information Technology* (iRAFIT2012) *Proceedings Published in International Journal of Computer Applications®* (IJCA), (2012), 21–25.
[10] A.D. Bakar, A. Sultan, H. Zulzalil and J. Din, "Predicting maintainability of object-oriented software using metric threshold." *Information Technology Journal*, 13 (2014) 1540-1547.
[11] S. Singh, M. Kaur, "Deriving and validating software metrics threshold values for design errors" *International Journal of Engineering Technology and Scientific Research* Volume 2 Issue 2, (2016).
[12] Kıral, A., Ayyıldız, T.E., "Comparison of software quality metrics: a case study." 2018, *12th Turkish National Software Engineering Symposium* (UYMS'18), Istanbul, TURKEY, 10-12 September, in press.
[13] K. Yamashite et al., "Thresholds for size and complexity metrics: a case study from the perspective of defect density," 2016 *IEEE International Conference on Software Quality, Reliability and Security* (QRS), Vienna, (2016), 191-201.
[14] G. Marcela et al. "A survey of metrics for UML class diagrams." *Journal of Object Technology* 4 (2005), 59-92.
[15] S. Demeyer, S. Ducasse and M. Lanza, "A hybrid reverse engineering approach combining metrics and program visualisation," *Sixth Working Conference on Reverse Engineering* (Cat. No.PR00303), Atlanta, GA, USA, (1999), 175-186.
[16] Mccabe.com, "All metrics thresholds in McCabe iq", [Online]. Available: http://www.mccabe.com/pdf/McCabe%20IQ%20Metrics.pdf. [Accesed: 7- Oct-2018].

Current Address: Alper KIRAL: Başkent University, Department of Computer Engineering, Ankara, 06790, Türkiye
E-mail: alperkiral@gmail.com
Orcid ID: https://orcid.org/ 0000-0002-4018-0419
Current Address: Tülin ERÇELEBİ AYYILDIZ: Başkent University, Department of Computer Engineering, Ankara, 06790, Türkiye
E-mail: ercelebi@baskent.edu.tr
Orcid ID: https://orcid.org/ 0000-0002-7372-0223