International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552 Volume- 7, Issue- 6, November 2019
https://doi.org/10.21276/ijircst.2019.7.6.4
www.ijircst.org

MODULARITY: A MAJOR ASPECT TO MAINTAINABILITY

Mohit Kumar, Dr. Abdullah, Dr. Jarnail Singh

ABSTRACT- Researchers and Practitioners advocated that modularity aspect of software is highly desirable and significant for developing quality oriented maintainable software. Despite the fact modularity is vital and highly significant aspect for software development process, it is poorly managed. This paper here highlights the importance of modularity broadly and also as an important contributor of software maintainability. In this paper a correlation between the major attributes of object oriented design and modularity has been ascertained. A modularity quantification model using multiple linear regression has been proposed for object oriented design. Finally, the validation of the proposed modularity quantification model is made known by means of experimental tests and the results show that the model is highly significant.

KEYWORDS- Maintainability, Modularity, Reusability, Testability, Design phase, Object Oriented Design.

I. INTRODUCTION

Modularity is an important key contributor to maintainability analysis and quantification for delivering high class testable and maintainable software. It constantly guides and supports to avoid wastage of resources as well as enable the designers for continuous improvement in the maintenance process [1, 2 3, 4, 24, 25]. Modularity has a direct relation to software maintainability and has a major role in providing high quality maintainable and trustworthy software. The concept of Modularity is a major factor when we design and develop software and its constituents. Developing programs and its constituent components with good modularity continually improves and simplifies test operations and maintenance during and after implementation [5, 6, 21, 22, 23,]. It encourages and supports improvement in software quality at design stage in the development of software. An accurate measure of software quality totally depends on maintainability quantification as a result estimating efforts in measuring maintainability is a complex problem attracting considerable research attention. Maintainability has always been an indefinable concept. Its truthful quantification or assessment is a complex exercise for the reason that of the various potential factors influencing maintainability [7, 8, 9, 10, 19, 20].

II. MAPPING BETWEEN MODULARITY AND OBJECT ORIENTED DESIGN PROPERTIES

According A wide-ranging assessment/review of object oriented design and development was showed in [11, 12, 13, 14, 15, 16, 17, 18, 26, 27], to develop a basis for mapping design properties to quality attribute modularity. In view of this fact, a relation figure is proposed between the major properties of object oriented design Modularity as shown in Fig. 1. The mapping puts in place a contextual impact relationship among Modularity and object oriented design properties and the related design metrics.

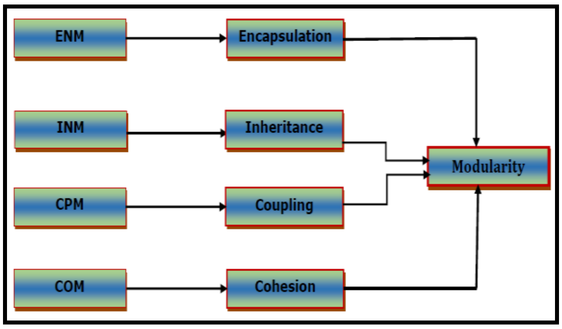

Fig. 1: Relation among Modularity and object oriented design properties and metrics

III. MODULARITY QUANTIFICATION MODEL (MQMOOD)

Measurement Grounded upon the relationship exposed in Fig.1, we propose a model for modularity quantification. Here we have implemented the concept of multiple linear regression (MLR) to develop a model for Modularity.

Manuscript received November 22, 2019

Mohit Kumar, Ph.D. (P), Department of Computer Science & Engineering, Sai Nath University, Ranchi Jharkhand, India,(mohitsky25@gmail.com)

Dr. Abdullah, Assistant Professor, Department of Information Technology, Adigrat University (A Public University), Adigrat Tigray, Ethiopia-Africa

Dr. Jarnail Singh, Professor University Institute of Computing, Chandigarh University, Panjab, India

Copyright © 2019. Innovative Research Publication. All Rights Reserve

Modularity=ß+ A1× Encapsulation **+ A2×** Inheritance **+ A3×** Coupling **+ A4×** Cohesion

Eq. (1)

We used SPSS to calculate the coefficients and the final modularity model that we arrived at is

Modularity = 8.961-.391× Encapsulation -2.044× Inheritance -1.608× Coupling +1.857× Cohesion

Eq. (2)

The datasets for developing and validating Modularity model is acquired from [28] that have been collected through the class diagrams. It includes a set of twenty (20) class diagrams along with the value of metrics of each of these. Along with this, we have the actual mean values of different ratings by experts of Software Modularity for these projects. These are called 'Known Value' here in this paper. Table 1 shows the coefficients for Modularity quantification model. We use the values we get from the unstandardized coefficients component of the table 1 to help develop the regression equation (2). The results of this trial experiment in assessment of modularity meet expectations and are very promising to attain maintainability index of object oriented design for small cost Software maintenance.

Table 1: Coefficients for Modularity Quantification Model

Coefficients[a]						
		Unstandardized Coefficients		Standardized Coefficients		
Model		B	Std. Error	Beta	t	Sig.
1	(Constant)	8.961	2.735		3.277	.022
	Encapsulation	-.391	.438	-.149	-.892	.413
	Inheritance	-2.044	1.248	-.263	-1.637	.163
	Coupling	-1.608	.467	-.549	-3.441	.018
	Cohesion	1.857	.477	.644	3.896	.011
a. Dependent Variable: Modularity						

The results of summarized model table 2 are useful when calculating multiple regression. The coefficient determinant (R) exhibits the strong relation between the independent variables and the respective dependent variable. The value of this coefficient when squared i.e. R (square) from the table depicts the coefficient of determination.

Table 2: Modularity Quantification Model Summary

Table 2: Modularity Quantification Model Summary

Model Summary									
					Change Statistics				
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	R Square Change	F Change	df1	df2	Sig. F Change
1	.937[a]	.879	.782	.59274	.879	9.071	4	5	.016
a. Predictors: (Constant), Cohesion, Coupling, Inheritance, Encapsulation									

IV. STATISTICAL SIGNIFICANCE BETWEEN MODULARITY AND OBJECT ORIENTED DESIGN PROPERTIES

The applications that are used in displaying the statistical significance among Modularity and object oriented design properties have been taken from [28] we categorized the applications as: System I, System J and System K. All the systems are commercial software projects implemented in C++ with the number of classes as shown in Table 3.

Copyright © 2019. Innovative Research Publication. All Rights Reserve

ISSN: 2347-5552 Volume- 7, Issue- 6, November 2019
https://doi.org/10.21276/ijircst.2019.7.6.4
www.ijircst.org

Table 3: Group and Projects for proposed MQM^{OOD}

Group Details	No. of Projects
System I	5
System J	5
System K	5

Table 4 gives the descriptive statistics for System I and Table 5 gives the correlation analysis for System I.

Table 4: Descriptive Statistics for System I

	Mean	Std. Deviation
Modularity	6.1000	3.33317
Encapsulation	3.9600	.82946
Inheritance	2.1600	1.70675
Coupling	3.0600	1.20540
Cohesion	2.3000	.51478

Table 5: Correlation Analysis for System I

	Modularity	Encapsulation	Inheritance	Coupling	Cohesion
Modularity	1	.819	.925*	.907*	.950*
Encapsulation	.819	1	.797	.571	.691
Inheritance	.925*	.797	1	.851	.982**
Coupling	.907*	.571	.851	1	.806
Cohesion	.950*	.691	.982**	.806	1
*. Correlation is significant at the 0.05 level (2-tailed).					
**. Correlation is significant at the 0.01 level (2-tailed).					

Table 6: Descriptive Statistics for System J

Descriptive Statistics		
	Mean	Std. Deviation
Modularity	6.1000	3.33317
Encapsulation	3.9200	.79498
Inheritance	2.0400	1.80499
Coupling	2.9000	1.23491
Cohesion	2.4600	.47223

Table 7: Correlation Analysis for System J

	Modularity	Encapsulation	Inheritance	Coupling	Cohesion
Modularity	1	.930*	.880*	.929*	.975**
Encapsulation	.930*	1	.890*	.820	.868
Inheritance	.880*	.890*	1	.895*	.753
Coupling	.929*	.820	.895*	1	.870
Cohesion	.975**	.868	.753	.870	1

Copyright © 2019. Innovative Research Publication. All Rights Reserve

	Modularity	Encapsulation	Inheritance	Coupling	Cohesion
Modularity	1	.930*	.880*	.929*	.975**
Encapsulation	.930*	1	.890*	.820	.868
Inheritance	.880*	.890*	1	.895*	.753
Coupling	.929*	.820	.895*	1	.870
Cohesion	.975**	.868	.753	.870	1
*. Correlation is significant at the 0.05 level (2-tailed).					
**. Correlation is significant at the 0.01 level (2-tailed).					

Table 8: Descriptive Statistics for System K

	Mean	Std. Deviation
Modularity	6.1000	3.33317
Encapsulation	4.0000	.65574
Inheritance	1.9600	1.87563
Coupling	2.8600	1.25817
Cohesion	2.4000	.47958

Table 9: Correlation Analysis for System K

	Modularity	Encapsulation	Inheritance	Coupling	Cohesion
Modularity	1	.906*	.868	.949*	.996**
Encapsulation	.906*	1	.990**	.927*	.882*
Inheritance	.868	.990**	1	.887*	.842
Coupling	.949*	.927*	.887*	1	.920*
Cohesion	.996**	.882*	.842	.920*	1
*. Correlation is significant at the 0.05 level (2-tailed).					
**. Correlation is significant at the 0.01 level (2-tailed).					

Table 10 summarizes the result of the correlation analysis for Modularity quantification model, which shows that for all the System, Encapsulation, Inheritance, Coupling, cohesion are highly correlated with modularity. The value of correlation 'r' lies between ±1, positive value of 'r' in Table 10, designates positive correlation between the two variables. The value of 'r' close to +1 specifies high degree of correlation between the two variables in above Table.

Table 10: Correlation Analysis Summary

	Modularity × Encapsulation	Modularity× Inheritance	Modularity × Coupling	Modularity × Cohesion
System I	.819	.925*	.907*	.950*
System J	.930*	.880*	.929*	.975**
System K	.906*	.868	.949*	.996**
*. Correlation is significant at the 0.05 level (2-tailed).				
**. Correlation is significant at the 0.01 level (2-tailed).				

V. EXPERIMENTAL VALIDATION OF MODULARITY QUANTIFICATION MODEL

The empirical validation is an important phase of proposed research. Empirical validation is the exact approach and practice to justify the model acceptance. This part of study focuses on the way the model proposed above is able to evaluate the Modularity calculated in object oriented software(s) at SDLC design stage. This experimental validation exists as a crucial step of proposed research to estimate Modularity quantification model for better and high level adaptability. Therefore with this objective a

Copyright © 2019. Innovative Research Publication. All Rights Reserve

ISSN: 2347-5552 Volume- 7, Issue- 6, November 2019
https://doi.org/10.21276/ijircst.2019.7.6.4
www.ijircst.org

validation of the Modularity quantification model and it is done using experimental tests.
In order to validate the developed Modularity quantification Model the projects viz. Pr1, Pr2, Pr3, Pr4, Pr5, Pr6, Pr7, Pr8, Pr9 and Pr10 were taken from [28] . The known Modularity rank of the provided projects class diagram is shown in Table 4.11.

Table 11: Known Modularity Value

Pr1	Pr2	Pr3	Pr4	Pr5	Pr6	Pr7	Pr8	Pr9	Pr10
7.8	6.9	7.2	7.4	7	8.1	8.5	9.1	8.9	9.3

Table 12: Known Modularity Rank

Pr1	Pr2	Pr3	Pr4	Pr5	Pr6	Pr7	Pr8	Pr9	Pr10
5	1	3	4	2	6	7	9	8	10

Using the similar set of data for the given projects class diagram Modularity was calculated using proposed Modularity quantification model and the results are shown in Table 13.

Table 13: Calculated Modularity Value Using Proposed Model MQM^{OOD}

Pr1	Pr2	Pr3	Pr4	Pr5	Pr6	Pr7	Pr8	Pr9	Pr10
2.2	2.0	3.5	3.3	3.5	4.2	3.9	6.6	8.0	8.1

Table 14: Calculate d Modularity Rank Using Proposed Model MQM^{OOD}

Pr1	Pr2	Pr3	Pr4	Pr5	Pr6	Pr7	Pr8	Pr9	Pr10
2	1	4	3	5	7	6	8	9	10

Charles Speraman's rank relation rs was used to test the significance of correlations calculated amidst Ranks of Modularity via proposed model and the ranks Known in it. The 'rs' was calculated using the formula given as under:

$$r_s = 1 - \frac{6\Sigma d^2}{n(n^2-1)} \quad -1.0 \leq rs \leq +1.0 \qquad \text{Eq. (3)}$$

'd' = difference that exists in Calculated Rank and Known Rank of Modularity.
'n' = total quantity of Projects taken in the experimentation.

Table 15: Computed Rank, Actual Rank and their Relation

No. of Projects	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Computed Ranks	2	1	4	3	5	7	6	8	9	10
Known Ranks	5	1	3	4	2	6	7	9	8	10
d^2	9	0	1	1	9	1	1	1	1	0
$\sum d^2$	24									
r_s Calculated(Values)	0.854545455									
$r_s > \pm.781$	✓									

The correlation value among calculated Modularity ranks using proposed model MQM^{OOD} and known ranks are shown in Table 4.15 above. Correlation value r_s undoubtedly show that the Modularity model is significant. The correlation meets the expectations standard showing high confidence, i.e. of 95%.

VI CONCLUSION

This paper shows the importance of Modularity and its correlation with object oriented design properties viz. encapsulation, coupling, inheritance and polymorphism. Modularity is one of the most significant factors for

Copyright © 2019. Innovative Research Publication. All Rights Reserve

evaluating maintainability of object oriented design. Using multiple linear regression formula on these attributes of object oriented design MQMOOD, a modularity quantification model is developed. The results obtained statically confirm the significance and acceptability of the proposed model. Modularity quantification model has been validated empirically via experimental test. The real-world validation of the Modularity model accomplishes that developed model is highly dependable, acceptable and significant. The paper concludes that there is a high correlation between Modularity and design properties.

REFERENCES

[1] Singh, Hardeep, and Aseem Kumar. "A Novel Approach to Enhance the Maintainability of Object Oriented Software Engineering During Component Based Software Engineering." International Journal of Computer Sci. and Mobile Computing 3.3 (2014): 778-786.

[2] Al Dallal, Jehad. "Object-oriented class maintainability prediction using internal quality attributes." Information and Software Technology 55.11 (2013): 2028-2048.

[3] 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

[4] P. Oman and J. Hagemeister, “Metrics for assessing a software system's maintainability,” Software Maintenance, 1992, pp. 337 - 344.

[5] K.K. Aggarwal, Yogesh Singh. New Age International, Jan 1, 2005 - Software engineering.

[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] Geoff R. Dromey's Model, (Feb 1995) (vol. 21 no. 2), IEEE Transaction on Software Engineering, A Model for Software Product Quality.

[8] S. Muthanna, K. Kontogiannis, K. Ponnambalaml and B. Stacey, “A Maintainability Model for Industrial Software Systems Using Design Level Metrics”, In Working Conference on Reverse Engineering (WCRE’00), 2000

[9] 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.

[10] 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.

[11] Abdullah, Dr, Reena Srivastava, and M. H. Khan. "Testability Measurement Framework: Design Phase Perspective”. International Journal of Advanced Research in Computer and Communication Engineering, Vol. 3, Issue 11, Pages 8573- 8576 November 2014

[12] 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.

[13] Abdullah, Dr, M. H. Khan, and Reena Srivastava. “Flexibility: A Key Factor To Testability”, International Journal of Software Engineering & Applications (IJSEA), Vol.6, No.1, January 2015. DOI: 10.5121/ijsea.2015.6108

[14] Abdullah, Dr, Reena Srivastava, and M. H. Khan. "Testability Estimation of Object Oriented Design: A Revisit". International Journal of Advanced Research in Computer and Communication Engineering, Vol. 2, Issue 8, pages 3086-3090, August 2013

[15] 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.

[16] Abdullah, Dr, Reena Srivastava, and M. H. Khan. “Modifiability: A Key Factor To Testability”, International Journal of Advanced Information Science and Technology, Vol. 26, No.26, Pages 62- 71 June 2014.

[17] Sub has Chandra Misra, “Modeling Design/Coding Factors That Drive Maintainability of Software Systems”, Software Quality Journal, 13, pages 297-320, 2005.

[18] Abdullah, Dr, M. H. Khan, and Reena Srivastava. “Testability Measurement Model for Object Oriented Design (TMMOOD)”. International Journal of Computer Science & Information Technology (IJCSIT), Vol. 7, No 1, February 2015, DOI: 10.5121/ijcsit.2015.7115.

[19] . B. Basili, L. Briand, and W. L. Melo, A validation of Object Oriented Metrics as Quality Indicators, IEEE Trans. Softwaree Engineering, Vol.22, No. 10 pp. 751 -761, Oct-1996.

[20] 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.

[21] Alisara Hincheeranan and Wanchai Rivepiboon,” A Maintainability Estimation Model and Tool.” International Journal of Computer and Communication Engineering, Vol. 1, No. 2, July 2012.

[22] 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.

[23] 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.

Copyright © 2019. Innovative Research Publication. All Rights Reserve

ISSN: 2347-5552 Volume- 7, Issue- 6, November 2019
https://doi.org/10.21276/ijircst.2019.7.6.4
www.ijircst.org

[24] Boehm, B. W., Brown, J. R., Kaspar, H., Lipow, M., McLeod, G., and Merritt, M., (1978) Characteristics of Software Quality, North Holland.
[25] ISO 9126-1 Software Engineering - Product Quality - Part 1: Quality Model, 2001.
[26] Sommerville, I. (1992). Software Engineering. 4th ed. New York, Addison- Wesley.
[27] Hordijk, Wiebe, and Roel Wieringa. "Surveying the factors that influence maintainability: research design." ACM SIGSOFT Software Engineering Notes. Vol. 30. No. 5. ACM, 2005.
[28] Mobo Dexter Software India Pvt. Ltd., Novel Tech Park, Third Floor, #43/4, GB playa, Hosur Road Bangalore.

Copyright © 2019. Innovative Research Publication. All Rights Reserve