International Journal of Control Theory and Applications

ISSN : 0974-5572

Volume 9 • Number 45 • 2016

Empirical Evaluation of Aspect Oriented Software Quality Model Using Mutli-Criteria Decision Making Based AHP Approach

Pradeep Kumar Singh[a] and Om Prakash Sangwan[b]

[a]*Assistant Professor (Senior Grade), Department of Computer Science and Engineering Jaypee University of Information Technology (JUIT) Waknaghat, Solan, HP*

E-mail: pradeep_84cs@yahoo.com

[b]*Associate Professor, Department of Computer Science and Engineering Guru Jambheshwar University of Science & Technology Hisar, Haryana, India*

E-mail: sangwan_op@yahoo.co.in

Abstract: This paper presents a new Aspect Oriented (AO) software quality model *i.e*. Proposed Aspect Oriented Software Quality Model (PAOSQMO) using important quality characteristics and their respective sub-characteristics. Security is exposed as a new characteristic along with its set of sub-characteristics by proposed quality model. PAOSQMO has derived Sub-characteristics for security as confidentiality, integrity, accountability, authenticity and non-repudiation. In order to validate the proposed model, we did the empirical analysis. Total fifty six participants from software companies, academics and research labs have been participated during our survey conducted to identify the missing attributes in AO software. A survey form was given to conduct the survey and collect the requirements of missing attributes in AO software. We have identified different attributes and considered the most frequent attributes for further analysis. Firstly, we have classified the collected attributes from survey into characteristics and sub-characteristics using experts responses. Thereafter, pair wise relative weights for each characteristic and their respective sub-characteristics are taken. Finally, we have applied one of the MCDM approach i.e. Analytical Hierarchical Process (AHP) on PAOSQMO for empirical evaluation of quality model by considering three Aspect Oriented Projects.

Keywords: *Software Quality, Quality Models, Aspect Oriented Software, Software Quality Attributes.*

1. INTRODUCTION

Several software quality models derived for the assessment of software applications. In this paper, we have reviewed number of software quality models for object oriented and aspect oriented software. We have identified the demand of various missing attributes like security, confidentiality, technical accessibility, integrity, completeness, helpfulness, accountability, authenticity, and non-repudiation in available quality models for AO software based on conducted survey. So, there is a strong requirement to find new characteristics which can meet the demand of abstraction types and features of AO software. In this paper we have proposed a new aspect oriented software quality model to meet the above mentioned requirements in term of missing quality attributes.

This paper is organised into four parts. First part covers introduction and demand of proposed quality model followed by related work and empirical evaluation using AHP in the second and third part of paper. In the fourth section, we have discussed the major findings of proposed quality model for AO software.

2. RELATED WORK

In this section, similar to the approached adopted by [7-9]Singh et *al.* for identifying the most relevant papers, we have also considered papers related to software quality in context to AO software. Various quality models suggested for Module Oriented and Object Oriented Paradigm. In 1977, [10]Mc Call *et al.* presented a quality model called McCall's Software Quality Model and it also named as Classical Quality Model. In 1987, it was later modified and changed as the MQ Model by Watts. Boehm *et al.* discussed one of the popular quality model called Boehm's Software Quality Model[11]. In the similar direction, FURPS quality model, Evans & Marciniak's quality model and Deutsch & Will's quality model (1998) were presented. Among all these quality models, FURPS became most popular because it was the first model based on industrial approach by Hewlett-Packard (HP). Later on, FURPS was extended to FURPS + by IBM to make it widely acceptable for the software industry.

Several software quality models were came into existence till 90's. Because of the availability of several software quality models it was difficult to select the quality models among software practitioners. Then, International Organization for Standardization/International Electro-technical Commission (ISO/IEC) has taken the initiative for standardization of quality models. Therefore, in 1991, ISO/IEC derived a quality model called ISO/IEC Quality Model, which changed to ISO/IEC 9126 Quality Model (ISO/IEC 9126, 2001-2003) and ISO 9126 considered as the part of the ISO 9000 standard. Dromey R.G. (1995) derived a software quality model by accumulating one characteristic into ISO/IEC 9126 Quality model and it is called as Dromey's Software Quality Model.

Most of the reported software quality models are related to either legacy software or object-oriented software. AOP software composed of class and aspect, so the quality assessment of AO software may not be made by the quality models for module and object oriented software. Hence, there is the basic difference between the class and aspect behaviour and structure (aspect have pointcut, advices and joinpoints), abstraction. Quality model for assessing the quality of AO projects need to be developed separately by considering AOP features. Since AO technology cannot stand on its own and aspectual modules are integrated with MO modules or with OO classes[12]. So, some of the software quality characteristics/sub-characteristics of ISO/IEC 9126 may be applicable to AO quality model.

One of the Aspect Oriented Software (AOS) quality model was presented by [13]Kumar *et al.* and they named their model as Aspect-Oriented Software Quality Model (AOSQUAMO). Reusability, Complexity, Code-Reducibility and Modularity were integrated as four sub-characteristics in their model while comparing to of ISO/IEC 9126 quality model.

In addition to the AO quality model by [13]Kumar *et al.*, [14]Castillo *et al.* presented another quality model for AO software. Author has taken common framework related to UML model and named as; REASQ (Requirements, Aspects and Software Quality) model. This model integrated the ontology for reasoning, handling, understanding and reuse of software using the Protégé Tool in AO software. Area of quality models for aspect oriented software is still not mature and quality attributes classification according to AOSD requires more standardization and validation. As AOP enhance cross cutting features like security, exceptional handling etc., quality attributes according to these features must be taken care for AO software quality model.

Because of new abstraction type in AO technology, some new software quality characteristics and sub-characteristics are required to be added, which may cover new features of AO technology. So, there is a strong requirement to derive Aspect-Oriented Quality Model similar to ISO/IEC 9126 in which security, confidentiality, technical accessibility, integrity, completeness, authenticity, and non-repudiation should be considered. So, in order to overcome these issues, we have Proposed an Aspect Oriented Software Quality Model (PAOSQMO).

3. EMPIRICAL EVALUATION OF PAOSQMO

All the seven characteristics with their total twenty nine sub-characteristics of our proposed quality model are shown in Table 1. However, while doing the comparison with existing quality models, we have explored that few of the characteristics and sub-characteristics of our proposed quality model are already available. Although, none of the study presented on software model taken security as a characteristics for AO software with best of our knowledge. In Table 1, all newly added characteristics and sub-characteristics of our proposed model are shown in **italics** for AO software.

Table 1
Proposed AO Software Quality Model (PAOSQMO)

Quality Type	*Characteristics*	*Sub Characteristics*
Software Product Quality	C1: Portability	SC11:Adaptability
		SC12:Transferability
		SC13:Installability
	C2: Performance	SC21:Time Behavior
		SC22:Resource Behavior
	C3: Maintainability	SC31: Reusability
		SC32: Analyzability
		SC33: Testability
		SC34:Changeability
		SC35: Stability
		SC36: Modularity
	C4: Security	SC41:Confidentiality
		SC42:Integrity
		SC43:Accountability
		SC44:Authenticity
		SC45:Non-repudation
	C5: Functionality	SC51: Suitability
		SC52:Accuracy
		SC53:Interoperability
		SC54:Completeness
		SC55: Co-existence
	C6: Reliability	SC61:Avilability
		SC62: Fault Tolerance
		SC63:Recoverability
	C7: Usability	SC71:Understandability
		SC72:Learnability
		SC73:Operatability
		SC74:Helpfulness
		SC75: Technical Accessibility

3.1. Assessment of PAOSQMO using Analytical Hierarchical Process

In this section we have used AHP to evaluate the quality of PAOSQMO. Consider m number of attributes to be mapped, C_1, C_2,……. C_m and represent the relative weight (or priority) of C_i with respect to C_j as a_{ij} and a square matrix $A = [a_{ij}]$ of order m[6].

Suppose for n number of factors, F_1, F_2….F_n are considered, which we are going to compare. Relative weight of F_i with respect to F_j is denoted as m_{ij} and a square matrix $A = [m_{ij}]$ of order n as given in equation (1) is derived.

$$A = [m_{ij}]$$

$$= \begin{matrix} \\ F_1 \\ F_2 \\ \cdot \\ F_n \end{matrix} \begin{pmatrix} F_1 & F_2 & \cdot & F_n \\ 1 & m_{12} & \cdot & m_{12} \\ 1/m_{12} & 1 & \cdot & m_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ 1/m_{1n} & 1/m_{2n} & \cdot & 1 \end{pmatrix} \quad (1)$$

Here $m_{ij} = 1/m_{ji}$, for $i \neq j$, and $m_{ii} = 1$ all i. This matrix is said to be reciprocal matrix. For a matrix involving human judgments, the judgments can be inconsistent to a greater or lesser degree[1]. In such cases, find vector ω satisfying the equation (1.2).

$$A\omega = \lambda_{max}\,\omega,$$

and

$$\lambda_{max} \geq n \quad (2)$$

Here ω is Eigen vector and λ_{max} represents Eigen values. The dissimilarity among λ_{max} and n, if any, is an indicator of inconsistency for the judgments. Saaty (1980) proposed a Consistency Index (CI) and Consistency Ratio (CR) to validate the steadiness of the comparison matrix. For validation following equations are defined as:

$$CI = \frac{\lambda \max - n}{n - 1} \quad (3)$$

$$CE = \frac{CI}{RI} \quad (4)$$

Here, RI is the average consistency index over many random entries of the same reciprocal matrix. Saaty (1980) suggested that if the CR is greater than 0.1, the set of judgments may be too inconsistent to be reliable. In such situation, a new comparison matrix is needed to be prepared until CR ≤ 0.1.

3.2 Allocating the weights to characteristics and sub-characteristics

In order to identify the characteristics and sub characteristics for aspect oriented software, we conducted a survey on 56 professionals. In these participants, ten of them were from software industries, sixteen of them from academics (University/ Colleges) and thirty of them were PG (M.Tech.(CSE)) students. All the participants were having research interest in aspect oriented software development and all selected PG students have gone through the Aspect Oriented Programming and Software Quality Assurance courses. We have prepared a detailed questionnaire and distribute it to all participants. In first phase the questions are designed in such a way, so that the main characteristics and sub-characteritics of aspect oriented software can be identified. Each participant is also encouraged to give individual remarks at the end of questioner too. Thereafter, the all participants were requested to weight the identified characteristics and their relative sub characteristics. A table is used to fill the pair wise relative weight value of seven characteristics from C_1 to C_7. The mean of the collected samples of pair wise relative weights are given in square matrix $A = [a_{ij}]$ of order seven in equation, which is derived using the equation (1.1) to apply AHP. Now, we have calculated the eigen values and eigen vectors to find the corresponding weight of C_1, C_2, C_3, C_4, C_5, C_6, C_7 and consistency ration (CR). We create a reciprocal matrix after that to calculate the eigen vector and values for CI and CR.

3.3. Allocating the Weights to Characteristics of PAOSQMO

We use the values from survey form and assign it to a square matrix. Firstly, we assign pair-wise relative weight values to all seven factors A_1 to A_7 using equation (1.1). Next step is to determine Eigen vector and Eigen values to get corresponding weights of A_1, A_2, A_3, A_4 , A_5 , A_6 , A_7 and consistency ratio (CR).

$$A = m_{ij}$$

$$= \begin{matrix} A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \\ A_6 \\ A_7 \end{matrix} \begin{pmatrix} A_1 & A_2 & A_3 & A_4 & A_5 & A_6 & A_7 \\ 1 & 0.333 & 0.111 & 0.333 & 0.111 & 0.142 & 0.20 \\ 3 & 1 & 0.111 & 1 & 0.111 & 0.333 & 0.142 \\ 9 & 9 & 1 & 9 & 1 & 5 & 3 \\ 3 & 1 & 0.111 & 1 & 0.111 & 0.11 & 0.20 \\ 9 & 9 & 1 & 9 & 1 & 3 & 5 \\ 7 & 3 & 0.2 & 9 & 0.333 & 1 & 1 \\ 5 & 7 & 0.33 & 5 & 0.20 & 1 & 1 \end{pmatrix} \quad (5)$$

3.4. Determining Eigen Vector and Eigen Values

There are many ways to find the Eigen vector. Multiply all the entries in each row of the matrix A and then take the n^{th} root (in our case 7^{th} root) of the product helps in getting eigen vector. The n^{th} root is summed and this sum is used to normalize the eigen vector elements. Table2 shows all the calculations and it is very clearly shown that A_3 *i.e.* Maintainability is found to be the most important factor in PAOSQMO with eigen vector as **0. 328826**. However, A_1 *i.e.* Portability is found to be least important one corresponding to eigen vector values as **0.021263**. In order to calculate **A.ω,** we multiply the matrix (A_1 to A_7) from eigen vector (**ω**). Calculation for the first row of Table 1.2 is shown below:

$$\begin{aligned} 1 \times 0.021263 \; & + \; 0.333 \text{ X } 0.035198 + 0.111 \times 0.328826 + 0.333 \times 0.03233 + 0.111 \times 0.3277 \\ & + \; 0.14285 \text{ X } 0.129372 + 0.2 \text{ X } 0.12527 \\ & = \; 0.160074 \end{aligned}$$

and the values for remaining six rows can be calculated in the similar way and estimated as 0.2650456, 2.478388, .24361298, 2.4701822, 0.974981621 and 0.944058858.

As given in equation (1.2), $A.\omega = \lambda_{max}.\omega$, and $\lambda_{max} \geq 7$, next step is to get the product of $A.\omega$. Eigen values λ_{max} can be evaluated by applying $\lambda_{max} = (A.\omega / \omega)$ and these seven λ_{max} values are calculated as 7.52833, 7.530089, 7.53706945, 7.534360077, 7.53708674, 7.53626457, 7.53619268. All these values of λ_{max} are greater than 7 which satisfy the condition of $\lambda_{max} \geq n$.

Now, we can evaluate CI using equation (3).

$$CI = \frac{\lambda \max - n}{n-1} = \frac{7.53419 - 7}{7-1} = 0.0890316 \quad (3)$$

Finally, the consistency ratio (CR) has been calculated for the set of judgments using CI for the considered samples. Value of RI can be taken from Table 1.3., it contains the upper row as the order of the random matrix and the lower row as the corresponding index of consistency[4-5].

Table 2
Eigen Vector and Eigen Value for main Factors

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	Eigen Vector(ω)	A.ω	$\lambda_{max} =$ A.ω / ω
A_1	1	0.33	0.11	0.33	0.11	0.142	0.20	0.021263	0.160074	7.52833
A_2	3	1	0.11	1	0.11	0.33	0.142	0.0351982	0.2650456	7.530089
A_3	9	9	1	9	1	5	3	0. 328826	2.478388	7.5370694
A_4	3	1	0.11	1	0.11	0.11	0.20	0.0323336	.24361298	7.5343600
A_5	9	9	1	9	1	3	5	0.325737	2.4701822	7.537086
A_6	7	3	0.20	9	0.33	1	1	0.129372	0.974981621	7.536264
A_7	5	7	0.33	5	0.20	1	1	0.12527	0.94405885	7.536192
Total								1.000		Mean =7.53419

Table 3
Satty Scales [2, 3, 4]

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49	1.51	1.48	1.56	1.57	1.59

Now, we can evaluate CR using equation (4)

$$CR = \frac{CI}{RI} = \frac{0.890316}{1.32} = 0.06744 \quad (4)$$

The calculated value of CR is 0.1, which indicates that the estimate is acceptable. So, we can say that our calculations are free from human bias and pass the essential conditions of AHP. Here A_3 *i.e.* maintainability has been estimated as the most important quality attribute in proposed model with value of **ω** as 0.328826. However, Portability is found to be least important one corresponding to eigen vector values as **0.021263.** Now we will assess the overall quality of AO software based on proposed model using below mentioned formula:

$$\text{AO Project Quality} = \sum_{i=1}^{n} \text{Comparative value of } SA_i * \text{Weight of } SA_i \quad (5)$$

SA_i signify sub-characteristic/ sub-attribute i. We have taken three aspect oriented projects from open source repositories and apply our proposed model to estimate the quality using above equation (1.6). All the calculation for the selected projects are shown in Table 4.

Table 4
Calculation of AO software quality

Charac-teristics	*Weights for Characteristics*	*Sub-Characteristics*	*Eigen vectors for sub-characteristics*	*Weights for Sub-characteristics*	*Compared Quality of Projects*		
					P1	*P2*	*P3*
A_1	0.021263	SC11	0.647	0.01375	0.003575	0.001375	0.0088
		SC12	0.253	0.00537	0.001289	0.000483	0.003598
		SC13	0.099	0.00210	0.000399	0.000168	0.001533
A_2	0.0351982	SC21	0.866	0.03047	0.018891	0.002742	0.008836
		SC22	0.133	0.00468	0.001217	0.000468	0.002995

Charac-teristics	*Weights for Characteristics*	*Sub-Characteristics*	*Eigen vectors for sub-characteristics*	*Weights for Sub-characteristics*	*Compared Quality of Projects*		
					P1	*P2*	*P3*
A_3	0.328826	SC31	0.02561	0.00841	0.00513	0.000757	0.002523
		SC32	0.04480	0.01473	0.003683	0.001473	0.009575
		SC33	0.38903	0.12790	0.024301	0.010232	0.093367
		SC34	0.07754	0.02549	0.006373	0.002549	0.016569
		SC35	0.24887	0.08180	0.054806	0.007362	0.019632
		SC36	0.21409	0.07036	0.018294	0.007036	0.04503
A_4	0.0323336	SC41	0.03055	0.00098	0.000657	0.000072	0.000171
		SC42	0.42034	0.01358	0.003531	0.001358	0.008691
		SC43	0.05339	0.00172	0.00043	0.000172	0.001118
		SC44	0.08767	0.00283	0.000764	0.00034	0.001726
		SC45	0.40802	0.01319	0.00831	0.000923	0.003957
A_5	0.327737	SC51	0.03079	0.01008	0.001915	0.000806	0.007358
		SC52	0.45026	0.14753	0.036883	0.014753	0.095895
		SC53	0.07626	0.02499	0.016743	0.002249	0.005998
		SC54	0.07733	0.02534	0.006082	0.002281	0.016978
		SC55	0.36533	0.11971	0.077812	0.00838	0.033519
A_6	0.129372	SC61	0.6545	0.08462	0.021155	0.008462	0.055003
		SC62	0.2759	0.03567	0.023899	0.00321	0.008561
		SC63	0.0695	0.00898	0.002155	0.000808	0.006017
A_7	0.12527	SC71	0.03184	0.00398	0.002507	0.000279	0.001194
		SC72	0.43533	0.05453	0.013087	0.004908	0.036535
		SC73	0.07542	0.00944	0.002549	0.001133	0.005758
		SC74	0.07591	0.00950	0.006365	0.000855	0.00228
		SC75	0.38147	0.04775	0.031038	0.003343	0.01337
	Quality values for AO Projects			0.9994	0.393837	0.088976	0.516586
	Ranking of the Assessed Projects				2	3	1

Based on the calculation shown in Table 4, we have estimated the overall quality values as 0.393837, 0.088976 and 0.516586 for all three considered projects. Using these computed values we ranked all three projects P1, P2, P3 as 2, 3 and 1 respectively.

4. DISCUSSIONS

Our proposed quality model for aspect oriented software i.e. PAOSQMO added *Security* as a new characteristic along with some of its sub-characteristics *i.e. confidentiality, integrity, accountability, authenticity and non-repudiation* while comparing to the existing software quality models like ISO 9126 and AOSQUAMO.

During analysis maintainability has been estimated as the most important quality attribute in proposed model with value of ω as 0.328826 and portability is found to be least important one corresponding to eigen vector as 0.021263. In addition to that during validation three open source projects were ranked based on the computed quality value for each project as 0.393837, 0.088976 and 0.516586 respectively for P3, P1 and P2 in descending order.

It is concluded that the proposed quality model for AO software can help in evaluating the quality of AO software and software professional can use this model to assess the aspect oriented software projects quality in quantitative way.

REFERENCES

[1] Ghosh S., Dubey S.K., Rana A. (2012). Object Oriented Software based on AHP, International Journal of Computer Science and Engineering, Vol. 4, No.11, pp. 1848-1853.

[2] Raju S. K., Kumar N. D. (2010). Multicriterion Analysis in Engineering and Management, PHI publication, pp. 103-107.

[3] Rønningen E. and Steinmoen T. (2004). Metrics for Aspect-Oriented Programming of middleware systems, Department of Computer and Information Science (IDI), Norwegian University (NTNU), Trondheim.

[4] Saaty T.L. (1990). How to Make a Decision: The Analytical Hierarchy Process, European Journal of Operation Research, Vol. 48, pp. 9-26.

[5] Saaty T.L. (2008). Decision Making with the Analytic Hierarchy Process, International Journal of Services Sciences, Vol. 1, No. 1, pp. 83-98.

[6] Zaerpour N., Rabbani M., Gharehgozli A. H., Tavakkoli-Moghaddam R. (2009). A Comprehensive Decision Making Structure for Partitioning of Make-to-Order, Make-to-Stock and Hybrid Products. Soft Computing, 13(11), pp. 1035-1054.

[7] Singh P.K., Panda R., Sangwan O.P. (2015). A Critical Analysis on Software Fault Prediction Techniques, World Applied Sciences Journal, Vol. 33 (3), pp. 371-379, 2015a.

[8] Singh P.K., Sangwan O.P., Panda R. (2014). A Study and Review on the Development of Mutation Testing Tools for Java and Aspect-J Programs, International Journal on Modern Education and Computer Science (IJMECS), Hong Kong. Vol. 6, No. 11, pp. 1-10.

[9] Singh P.K., Agarwal D., Gupta A. (2015b). A Systematic Review on Software Defect Prediction, Proceedings of the 9th INDIACom-2015, IEEE International Conference on Computing for Sustainable Global Development, IEEE Xplore, 11th - 13th March, 2015.

[10] McCall J. A., Richards P. K. and Walters G. F. (1977). Factors in Software Quality, Griffiths Air Force Base, N.Y. Rome Air Development Center Air Force Systems Command.

[11] Boehm B.W., Brown J. R. and Lipow M. (1976). Quantitative Evaluation of Software Quality. In IEEE Computer Society Press, pp. 592 - 605.

[12] Kiczales G., Lamping J., Mendhekar A., Maeda C., Lopes C., Loingtier J. M., and Irwin J. (1997). Aspect-Oriented Programming, in Proceedings of the 11th European Conference on Object-Oriented Programming, LNCS, Vol.1241, Springer-Verlag, pp. 220-242.

[13] Kumar A., Grover P. S. and Kumar R. (2009). A Quantitative Evaluation of Aspect-Oriented Software Quality Model, in ACM SIGSOFT Software Engineering Notes Volume 34, Number 5, pp. 1 - 9.

[14] Castillo I., Losavio F., Matteo A. and Boegh J. (2010). Requirements, Aspects and Software Quality: the REASQ model. In Journal of Object Technology, Vol. 9, No. 4, pp. 69-91.