International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-1, October, 2019

Software Component Quality Model

Mohamed Abdullahi Ali, Ng KengYap

Abstract: ***In Component Based Software Development (CBSD), applications are built from existing components either by assembling or replacing software parts. Reusing components may lead to faster software development and subsequently reduce cost and provide higher product quality. In CBSD, software component models define what components are and how they compose. However, no research has been done to assess the quality of software component models, to assess the characteristics of software component design. This paper proposed a software component quality model specifically to answer the question what characteristics make good component. A Systematic Literature Review (SLR) has been conducted by defining a robust protocol that combines automatic searches from different sources. The finding of the SLR has contributed to the development of quality model for CBSD, i.e. a proposed component quality model with metrics which is specific to software component design.***

Keywords: Software Component, Quality Model, Metrics

I.INTRODUCTION

Software engineering is a discipline that concerns with all aspects of software development including methodologies, project management and tools (Sommervile, 2010). Traditional software development approaches advocate phase by phase software process that lead to late delivery and cost overrun(Simão&Belchior, 2003). To tackle this, CBSD has arisen (Tiwari&Chakhraborty, 2015). CBSD is a software development approach that solves the problems in traditional software development by composing existing Components instead of starting development from scratch (Simão&Belchior, 2003;Tiwari&Chakhraborty, 2015). A challengefaced CBSD that is quality assurance of the underlying component which will eventually give to the quality of final product (Kaur& Singh, 2008). Software quality is “the degree to which system, system component or process meets specified requirements” (IEEE, 1990).Software component quality assurance increasingly important activity to ensure reliability in reused software components (Tiwari&Chakhraborty, 2015).

Revised Manuscript Received on October 30, 2019.
* Correspondence Author
Mohamed Abdullahi Ali*, Dept. of Software Engineering and Information System, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia
Ng KengYap, Dept. of Software Engineering and Information System, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia

Several software quality models were proposed but they are not applicable in CBSDfor example, internal structure in some quality models includes source code which may not be available in CBSD(Goulão, 2011). Hence, component quality models were proposed (Tiwari&Chakhraborty, 2015;Alvaro et al, 2006;Upadhyay et al, 2011; Bertoa&Vallecillo, 2002).However, none of them are specific to component design, because those quality modelsdetermined attributes including: functionality, efficiency, and reliability. However, this shows the need fora component quality model that is specific to design: is the most crucial phase in CBSD that allows us to detect poor design at earlier (Irwanto, 2010).

The contribution of this paper is to propose a software component quality model that is specific to component design.SLR has been conducted to identify the quality characteristics for good component and metrics.

The remainder of the paper is structured as follows: Section 2reviews existing research and related work. Section 3 discusses methodology. Section 4 presents the proposed software component quality model. Section 5discusses conclusion. Section 6 shows Acknowledgment. Finally, Section 7presents the references.

II.LITERATURE REVIEW

Existing Software Quality Models

Many software quality models were proposed to assure general software quality (Suman&Rohtak, 2014), such as (Grady, 1992; Boehm et al,1978; McCall et al, 1978). They are not software component specific to address characteristics such as the black-box nature of software components(Goulão, 2011).

Existing Component Quality Models

Several component quality models were proposed (Upadhyay et al, 2011; Bertoa&Vallecillo, 2002;Alvaro et al, 2005; Rawashdeh&Matalkah, 2006), they are nevertheless general component quality models. None of the quality models had been focusing on component design that determines the quality of the final product (Irwanto, 2010). Some existing component quality models that have been proposed by other researchers:

Software Component Quality Model (SCQM)

SCQM (Upadhyay, 2011) consists of eight component quality attributes. However, this model contains only a few attributes on component design including portability, reusability and usability.

Bertoa's Quality Model

Bertoa's quality model (Bertoa&Vallecillo, 2002) is a Component quality model that consists of several attributes for COTS. Therefore, this model contains only a few attributes on component design including compatibility, complexity and portability.

Alvaro Quality Model

Alvaro's quality model (Alvaro et al, 2005), also similar to Bertoa's model but provides component specific sub quality attributes. Hence, this model contains a few attributes on component design including configurability and self-containment.

III.METHODOLOGY

In order to investigate the characteristics and metrics influencing good component design, a SLR (Kitchen ham et al, 2010) has been conducted in 3 phases namely: planning, execution and reporting.

a. Planning: Includes:research questions, search string, literature resources, and inclusion and exclusion.

1. Research Questions: Two Research Questions (RQ) has been developed, namely: RQ1) 'what are the attributes for a good component design?' and RQ2) 'what are the metrics for a good component design?'.

2. Search String: The search string in figure 1 has been developed in order fordatabase search.

> (Quality attributes for component based software OR design quality characteristics for component based software) AND (design quality model for component based software OR quality model for software component design OR design quality model for component based software) AND (design metrics for component based software)

Fig. 1 Search String

Literature Resources: This review referred four electronic data bases which were namely: Springer Link, Google Scholar, IEEE Explore, and ACM.

Inclusion and exclusion: inclusion was all papers related to software component quality models which were published between 2000-2019, written in English. The exclusioncriteria, out of the period range and not written in English.

b. Execution: digital search was performed by using the developed search string. A total of 422 references remained. Next stage, title and abstract were scanned to evaluate the papers. Next stage, duplicate and irrelevant were rejected and 40 relevant papers were found. Finally, 20 papers were accepted for data synthesis of evidence.

c. Reporting: includes data synthesis and result. Data synthesis has been done for extracting information and addressing the answer to the research questions. Result, reported the component design attributes and metrics by using two RQ developed above. Table 1, presents the answer for RQ1.11 attributes were found for good component design.

Table. 1 What are the attributes for good component design?

No.	Metrics	Definition	No	Metrics	Definition
1	Interaction %age Metricsfor Component Integration (I%MCI)	Is a metric that measurescomponent integration by looking interaction density between components in a software system (Latika, 2011). I%MCI = Incoming interactions (Ii) + Outgoing interactions (Io)/Component Interactions (CI). CI is a ratio of available number ofincoming interaction to the outgoing interaction multiply 100%.	6	Cohesion of Methods within Component (COMC)	Related to the relatedness of methods in the component (Yadav&Tomar, 2014). COMC= COM (Counts of number of Methods same type)/TM (Total number of Methods).

2	Interface Method Complexity Metric for Black-Box (IMCM(BB))	Is a metric which enables to determine the interface complexity of the component, reusability and also use to measure testability in terms of interface complexity (Kaur& Singh, 2013a). IMCM(BB)= Wr + PCM(M). Wr stands the weight assigned to the category of return value's data type. PCM(M) is a parameter complexity metric that measures the of parameters method complexity.	7	Provided Services Utilization (PSU)	It's a metric that assess whether component has unnecessary interfaces (Van Der Hoek et al, 2004). Pactual refers number of services that component provides which are actually used by another component. Ptotal refers number services provided by component. PSU= Pactual/ Ptotal.
3	Coupling Complexity for Black-box Component (CCBC)	Is a metric that determines the coupling among components in terms of component interfaces (Kumar et al, 2014).IIc is the incoming interface which means required interfaces. OIc is the outgoing interfaces which means provided interfaces. CCBC= IIc + OIc.	8	Reference Parameter Density (RPD)	Can calculate Interface Reference Parameter Count (IRPC)/Interface Parameter Count (IPC) (Tonella et al, 1997). IRPC counts reference type parameters of all public methods in the interface of component. IPC is a total reference type method in the interface of the component.
4	Rate Component Customizability (RCC(c))	Is a configurability metric which determines writable properties in the interface of the provided component (Washisaki et al, 2003). RCC= Pw(c) is a number of writable properties in component divided by A(c) is the number of façade component.	9	Average Interface Complexity (AIC)	Is a metric that utilize to measure usability in terms of interface complexity (Agarwal et al, 2017). AIC= incoming interactions + outgoing interactions.
5	Rate Component Observability (RCO(c))	Is an encapsulation metric which determines readable properties in the interface of the provided component (Washisaki et al, 2003). RCO= Pr(c) is a number of readable properties of the component divided by A(c) which is the number of façade component.	10	Component Complexity Metric (CCM)	Is a metric that determines for component complexity (Kaur& Singh, 2013b). CCM= Component Coupling Complexity Metric (CCCM) + Interface Method Complexity Metric (IMCM).

Table 2, shows the answer for RQ2. 10 metrics were found for a good component design.

Table. 2 What are the metrics for good component design?

No	Attributes	Definition	No	Attributes	Definition
1	Compositionality	Describes the ability for composing together software componentsthrough well-defined interface (Ghani et al, 2016).	7	Testability	Is the ability in which easily observable the incoming and outgoing component interfaces (Simão&Bechior, 2003). The less interfaces the more testable.
2	Reusability	Express the degree how easily can assemble prefabricated components(Lau et al, 2005).	8	Cohesion	Express the ability in which functions performed by component are related (Gui& Scott, 2009).

3	Coupling	Is the degree of interdependenceamong components (Gui& Scott, 2009).	9	Usability	Describes the ability of a software component to be learned and understood (Alvaro et al, 2006).
4	Configurability	Is a degree that a component allowsto configure their behavior by small effort of user (Jack, 1998).	10	Slim	Is the degree in which component filled with necessary interfaces (McGrenere& Moore, 2000).
5	Encapsulation	Is the degree of controlling data access (Lua& Simone Di cola, 2017). Protect for data inside component to access outside.	11	Interface Documentation	Provides a precise documentation to the developers that concerns complete description for component's inputs, component's outputs (Parnas, 2006).
6	Complexity	Express the degree to which components hashuge interfaceslinked together that leadsdifficult to understand(Kaur& Singh A., 2013).			

IV.PROPOSED SOFTWARE COMPONENT QUALITY MODEL

The proposed software component quality model contains a set of quality attributes and metrics which are specific to component design. It contains a set of 11 attributes and 10 metrics which werefound from the conducted SLR.

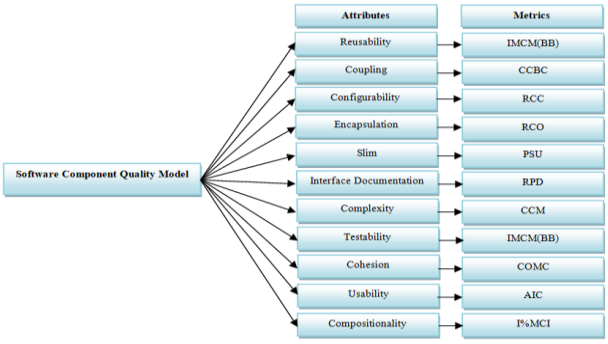

Fig. 2 Proposed Software Component Quality Model

V.CONCLUSION

As a number of CBSD systems being built continues to increase. Consequently, the need for a model that ensures quality characteristics of such systems becomes a necessity. The existing component quality models are not specific to CBSD in design phase. Therefore, this paper has proposed a new component quality model specific to design phase. Our proposed quality model will serve as a model for addressing design issues and producing high quality component. As future work, need to add additional quality attributes and metrics on the proposed software component quality model. Also, need to categorise main and sub attributes.

ACKNOWLEDGMENT

The authors of this work gratefully acknowledge by Ministry of Education (MOE), Malaysiafor financial support underFundamental Research Grant Scheme(FRGS) (Ref: FRGS/1/2015/ICT01/UPM/02/6).

REFERENCES

1. Agarwal, J., Dubey, S. K., & Tiwari, R. (2017). A Roadmap to Identify Complexity Metrics for Measuring Usability of Component-Based Software System. In Advances in Computer and Computational Sciences (pp. 33-41).Springer, Singapore.
2. Alvaro, A., Almeida, E. S., &Meira, S. L. (2005). Quality attributes for a component quality model. 10th WCOP/19th ECCOP, Glasgow, Scotland, 31-37.
3. Alvaro, A., De Almeida, E. S., &Meira, S. L. (2006, August). A software component quality model: A preliminary evaluation. In Software Engineering and Advanced Applications, 2006. SEAA'06. 32nd EUROMICRO Conference on (pp. 28-37). IEEE.
4. Bertoa, M. F., &Vallecillo, A. (2002). Quality attributes for COTS components.
5. Boehm, B. W., Brown, J. R., Kaspar, H., Lipow, M., & MacLeod, G. (1978). Merritt.: Characteristics of Software Quality.
6. Di Cola, S. Catch Me If You Can: To Use a Component You Need to Find It First.
7. Eric M. Dashofy (2002). Interoperability [PowerPoint Slides].Retrieved fromhttps://www.ics.uci.edu/~taylor/ICS221/slides/Interoperability.ppt.
8. Gill, Nasib S. "Reusability issues in component-based development." ACM SIGSOFT Software Engineering Notes28.4 (2003): 4-4.
9. Goulão, M. (2011). An overview of metrics-based approaches to support software components reusability assessment. arXiv preprint arXiv:1109.6802.
10. Gui, G., & Scott, P. D. (2009). Measuring Software Component Reusability by Coupling and Cohesion Metrics. JCP, 4(9), 797-805.
11. Grady, R. B. (1992). Practical software metrics for project management and process improvement. Prentice-Hall, Inc..
12. Ghani, N., Hedges, J., Winschel, V., & Zahn, P. (2016). Compositional game theory. arXiv preprint arXiv:1603.04641.
13. IEEE Standards Coordinating Committee.(1990). IEEE Standard Glossary of Software Engineering Terminology (IEEE Std 610.12-1990). Los Alamitos. CA: IEEE Computer Society, 169.

14. Irwanto, D. (2010, December). Visual Indicator Component Software to Show Component Design Quality and Characteristic.In Advances in Computing, Control and Telecommunication Technologies (ACT), 2010 Second International Conference on (pp. 50-54).IEEE.
15. Jack.(1998).Configurability.https://www.thwink.org/soft/article/future_app_dev/Configurability.html. accessed 4/8/2018.
16. Kaur, K., & Singh, H. (2008, July). A Metrics Based Approach to Evaluate Design of Software Components. In 18th ECOOP Doctoral Symposium and PhD Student Workshop (p. 17).
17. Kaur, K., & Singh, H. (2009). Evaluating an evolving software component: case of internal design. ACM SIGSOFT Software Engineering Notes, 34(4), 1-4.
18. Kaur, N., & Singh, A. (2013a). A Metric for Accessing Black Box Component Reusability. International Journal of Scientific & Engineering Research, 4(7), 1114-1121.
19. Kaur, N., & Singh, A. (2013b). A complexity metric for black box components. International Journal of soft computing and engineering, 3(2).
20. Kitchenham, B., Pretorius, R., Budgen, D., Brereton, O. P., Turner, M., Niazi, M., & Linkman, S. (2010). Systematic literature reviews in software engineering–a tertiary study. Information and Software Technology, 52(8), 792-805.
21. Kumar, S., Tomar, P., Nagar, R., &Yadav, S. (2014). Coupling metric to measure the complexity of component based software through interfaces. International Journal, 4(4).
22. Lau, K.K.,and Simone dicola (2017). An introduction to component based software development. world scientific.
23. Lau, K. K., Elizondo, P. V., & Wang, Z. (2005, May). Exogenous connectors for software components.In International Symposium on Component-Based Software Engineering (pp. 90-106).Springer, Berlin, Heidelberg.
24. Latika, M. (2011). Software component complexity measurement through proposed integration metrics. Journal of Global Research in Computer Science, 2(6), 13-15.
25. McCall, J. A., Richards, P. K., & Walters, G. F. (1977). Factors in software quality, volumes I, II, and III.US Rome Air Development Center Reports, US Department of Commerce, USA.
26. McGrenere, J., & Moore, G. (2000, May). Are we all in the same" bloat"?. In Graphics interface (Vol. 2000, pp. 187-196).
27. Parnas, D. L. (2006). Component Interface Documentation: What do we Need and Why do we Need it?. FRONTIERS IN ARTIFICIAL INTELLIGENCE AND APPLICATIONS, 147, 3.
28. Rawashdeh, A., &Matalkah, B. (2006).A new software quality model for evaluating COTS components. Journal of Computer Science, 2(4), 373-381.
29. Simão, R. P., &Belchior, A. D. (2003). Quality characteristics for software components: Hierarchy and quality guides. In Component-based software quality (pp. 184-206).Springer, Berlin, Heidelberg.
30. Sommerville, I. (2010). Software engineering. New York: Addison-Wesley.
31. Suman, M. W., &Rohtak, M. D. U. (2014). A comparative study of software quality models. International Journal of Computer Science and Information Technologies, 5(4), 5634-5638.
32. Scheller, T., & Kuhn, E. (2011, August). Measurable concepts for the usability of software components. In Software Engineering and Advanced Applications (SEAA), 2011 37th EUROMICRO Conference on (pp. 129-133). IEEE.
33. Tiwari, A., &Chakraborty, P. S. (2015, February). Software component quality characteristics model for component based software engineering. In Computational Intelligence & Communication Technology (CICT), 2015 IEEE International Conference on (pp. 47-51). IEEE.
34. Tonella, Paolo, GiulianoAntoniol, Roberto Fiutem, and Ettore Merlo. "Points to analysis for program understanding." In Proceedings Fifth International 1997.IWPC'97.Proceedings., Fifth Iternational Workshop on (pp. 90-99). IEEE..
35. Upadhyay, N., Despande, B. M., &Agrawal, V. P. (2011, January). Towards a software component quality model. In International Conference on Computer Science and Information Technology (pp. 398-412).Springer, Berlin, Heidelberg.
36. Van Der Hoek, A., Dincel, E., &Medvidovic, N. (2004, September). Using service utilization metrics to assess the structure of product line architectures. In Proceedings. 5th International Workshop on Enterprise Networking and Computing in Healthcare Industry (IEEE Cat. No. 03EX717)(pp. 298-308). IEEE.
37. Washizaki, H., Yamamoto, H., & Fukazawa, Y. (2003, September). A metrics suite for measuring reusability of software components.In Software Metrics Symposium, 2003.Proceedings. Ninth International (pp. 211-223). IEEE.
38. Yadav, K., &Tomar, P. (2014). Design of Metrics for Component-Based Software System at Design Level. International Journal of Engineering and Technical Research, 2(4), 285-289.