Software Component Quality Model

Mohamed Abdullahi Ali, Ng Keng Yap

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).

Several software quality models were proposed but they are not applicable in CBSD for 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 model determined attributes including: functionality, efficiency, and reliability. However, this shows the need for a 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 2 reviews existing research and related work. Section 3 discusses methodology. Section 4 presents the proposed software component quality model. Section 5 discusses conclusion. Section 6 shows Acknowledgment. Finally, Section 7 presents the references.

II. LITERATURE REVIEW

Existing Software Quality Models

Many software quality models were proposed to assure general software quality (Suman&Rothak, 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

Revised Manuscript Received on September 22, 2019.

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 Keng Yap, Dept. of Software Engineering and Information System, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia

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

Published By:
Blue Eyes Intelligence Engineering & Sciences Publication

Retrieval Number: A26591091192019@BEIESP
DOI: 10.35940/Ijeat.A2659.109119
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 (Kitchenham 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 for database search.

**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.

**Fig. 1 Search String**

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

<table>
<thead>
<tr>
<th>No.</th>
<th>Metrics</th>
<th>Definition</th>
<th>No.</th>
<th>Metrics</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Interaction %age Metrics for Component Integration (I%MCI)</td>
<td>Is a metric that measures component 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 of incoming interaction to the outgoing interaction multiply 100%.</td>
<td>6</td>
<td>Cohesion of Methods within Component (COMC)</td>
<td>Related to the relatedness of methods in the component (Yadav&amp;Tomar, 2014). COMC= COM (Counts of number of Methods same type)/TM (Total number of Methods).</td>
</tr>
</tbody>
</table>
Table 2 shows the answer for RQ2. 10 metrics were found for a good component design.

<table>
<thead>
<tr>
<th>No</th>
<th>Attributes</th>
<th>Definition</th>
<th>No</th>
<th>Attributes</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Compositionality</td>
<td>Describes the ability for composing together software components through well-defined interface (Ghani et al, 2016).</td>
<td>7</td>
<td>Testability</td>
<td>Is the ability in which easily observable the incoming and outgoing component interfaces (Simão &amp; Bechior, 2003. The less interfaces the more testable.</td>
</tr>
</tbody>
</table>


Software Component Quality Model

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coupling</td>
<td>Is the degree of interdependence among components (Guì &amp; Scott, 2009).</td>
</tr>
<tr>
<td>Configurability</td>
<td>Is a degree that a component allows to configure their behavior by small effort of user (Jack, 1998).</td>
</tr>
<tr>
<td>Encapsulation</td>
<td>Is the degree of controlling data access (Luá &amp; Simone Di cola, 2017). Protect for data inside component to access outside.</td>
</tr>
<tr>
<td>Complexity</td>
<td>Express the degree to which components hashuge interfaces linked together that leads difficult to understand (Kaur &amp; Singh A., 2013).</td>
</tr>
</tbody>
</table>

** 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 were found 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), Malaysia for financial support under Fundamental Research Grant Scheme (FRGS) (Ref: FRGS/1/2015/ICT01/UPM/02/6).

**REFERENCES**


