Software Metrics for Reusability of Component Based Software System: A Review

Jyoti Aggarwal[1] and Manoj Kumar[2]
[1]Department of Computer Science (ASET), Amity University, Noida
[2]School of Computer Science, University of Petroleum and Energy Studies, Dehradun

Abstract: *Component Based Software System (CBSS) have become most generalized and popular approach for developing reusable software applications. A software component has different important factors, but reusability is the most citing factor of any software component. Software components can be reused for the development of another software application, which further reduces the amount of time and effort of software development process. With the increase in the number of software components, requirement for identification of software metrics also increased for quantitative analysis of different aspects of components. Reusability depends on different factors and these factors have different impact on the reusability of software components. In this paper, study has been performed to identify the major reusability factors and software metrics for measuring those factors. From this research work, it will become easier to measure the reusability of software components, and software developers would be able to measure the degree of various features of any application which can be reused for developing other software applications. In this way, it would be easy and convenient to identify and compare the reusable software components and they could be reused in effective and efficient manner*.

Keywords: *Reusability metrics, software components, factors, software.*

Received January 21, 2020; accepted December 10, 2020
https://doi.org/10.34028/iajit/18/3/8

1. Introduction

Component based software development can be considered as both subsets and extensions of the current software development processes [9]. Latest software applications are large, complex, and packaged with variety of features. These important features (components) can be reused in some other applications also. Component based software engineering is the field, which work to identify these components from the software applications and allow the software developers to reuse these components to develop Component Based Software System (CBSS). Components available for reuse are already tested, experimented, and debugged. A new system developed using existing components turns out to be less expensive and almost pre-tested and debugged [18]. CBSS also reduces development cost and testing effort and also have many advantages like flexibility, reusability, functionality, higher productivity etc. These factors motivate the software industries to work on quality of CBSS rather than traditional software system. From a survey, it was identified that out of 118 companies around the world, approximate 53% of the organizations are using component-based approach for developing the software applications [1]. The reason for the same is the reusability of software components. Development cost of software applications can be reduced up to 20% by adapting the reuse strategy [10]. Reusability of software component improves the productivity and has positive impact on the software quality and maintainability. Reusable components should be adaptable, brief, consistence, flexible, simple, tested several times and reduces the error chances [19]. A component developed in any language may be available for reusing in different operating environment. For making any component reusable its size should also not be too large. These factors cause a big hurdle for component reuse. So, it becomes very important to follow some guidelines to make the component reusable. Selection of suitable reusability factors and metrics are also the part of those guidelines [14]. Software components can be reused in two ways: without any changes in the existing components and by doing some changes in the existing components [27]. Reusing the software with some changes is a difficult task as the developers must identify that where the changes are required and after those changes, complete component need to be tested again. The primary concern for selecting the appropriate reusable component is to measure the reusability of different components. Reusability is the non-functional aspect of the software, which is difficult to measure. For this purpose, software metrics are used. Software metrics is a method to quantify the attributes of a software process, product, and projects [7]. Software metrics automate the extraction of reusable components from the existing software system [4]. Software metrics are also required to estimate the reuse benefits and to provide easily interpretable and understandable results to the stakeholders [12]. It has been analyzed that different researchers have used different factors for working on reusability of CBSS and proposed different metrics for assessing those factors, but there is not a standard set of metrics for measuring the reusability of CBSS. The objective of this work is to study and analyze the exiting research work to identify the major reusability factors and software metrics for CBSS. By using these metrics, it would be possible to measure the reusability of different components which would be helpful for the software developers and researchers to select the most reusable software components for developing new software applications.

The paper is structured into four different sections: section 2 deals with the review process and study of related work is given in section 3. Finding and analysis of literature review is explained in section 4, followed by conclusion and future scope in section 5.

2. Review Process

The standard guidelines of review process [17] are:

2.1. Research Questions

- *RQ*1: Why existing software metrics are not sufficient for measuring reusability of CBSS?

Reason: There are so many reusability metrics or traditional and object-oriented software system, but these metrics are not sufficient for measuring reusability of CBSS. It is important to identify the reason that why different metrics are required?

- *RQ*2: What are the major reusability factors for CBSS?

Reason: Reusability depends on various factors. and each factor has its own importance. So, it is important to summarize these reusability factors at on place so that in future researchers can easily identify these factors.

- *RQ*3: What are the different software metrics for measuring different factors of reusability for CBSS?

Reason: For assessing these reusability factors, software metrics are required.

2.2. Search Process

Research papers were searched on different electronic databases and following search string was used:

(“software reusability” OR “metrics” OR “Component Based Software” OR “reusability assessment” OR “reusability factors”)

2.3. Inclusion and Exclusion Criteria of Research Papers

This work includes the existing work which focus on the importance of reusability, reusability factors and software metrics for reusability. Research papers, which were not written in English and not giving significant contribution for the objective of this paper were excluded.

2.4. Quality Assessment and Data Extraction

Quality assessment questions [17] were satisfied to assess the quality of included research work and useful information like reusability factors, software metrics, source of publication etc. are extracted.

3. Related Work

This section provides the review of existing work done in reusability of CBSS.

Aris and Salim [3] have reviewed various component models of research and industrial community to show its importance that how it helps to select the appropriate components for any application. The benefits of reusing the software components were explained in terms of effort, cost and productivity and software metrics were also used to calculate the reuse cost of the existing components [22]. Upadhyay *et al*. [28] have proposed a quality model for discussing about the major factors of CBSS. Reusability was also one of those factors which was further divided into two different sub-factors. Generality was considered as the important factor of reusability by Kalaimagal and Srinivasan [15]. An improved quality model was given to discuss about the configurability, generality, and understandability as the major reusability factors [16]. Rotaru and Dobre [23] have also done research on reusability factors and proposed that adaptability, composability, and interface complexity gave clear vision of reusability of any software system and for measuring these factors, metrics were also given by them. Sagar *et al*. [24] discussed that customizability, portability, documentation quality and interface complexity should be measured to assess the reusability of the component. Nyasente *et al*. [20] explained that maintainability, portability, documentation quality, generality and understandability play vital role for reusability of the object-oriented components and discusses the metrics to assess these factors. As per Hristov *et al*. [11], reusability can be measured in terms of availability, maintainability, adaptability, documentation, complexity, quality and cost of the components. In 1991, four metrics volume, operators, operand and cyclometric complexity are proposed for reusability and based on the value of these metrics reuse frequency is also calculated for different case studies [4]. Dumke and Schmietendorf [8] have proposed that number of public methods can be used for assessing the reusability. Component itself Reusability (CR) and Component Reuse Level (CRL) metrics were also proposed to assess the reusability [6]. One more complexity metrics based on number of interfaces, methods and variables are used for assessing the reusability [26]. Washizaki *et al*. [29] have used understandability, portability, and adaptability as major factors of reusability. A review has been done to identify the direct or indirect software metrics for measuring the reusability of black box and white box components [1]. Sharma *et al*. [25] have also performed a survey to identify the different aspects and metrics of reusability. Jha and Mishra [14] have reviewed some important aspects of CBSS and provide some guidelines to improve the reusability of CBSS. They have also discussed that how the most eligible component can be selected from a component repository. Padhy *et al*. [21] have proposed reusability metrics for partially adaptable, completely changeable and moderator capable components. Machine learning algorithm has been also applied to different reusability metrics for assessing the reusability of open-source software and 98.64% accuracy was achieved. A set of four metrics was also used to measure the reusability of CBSS. For this purpose, different components were used from different repositories [13].

From this literature review, it has been analyzed that ample research has been done on reusability of CBSS. For this purpose, different factors, metrics, and approaches are applied. It has been also observed that object-oriented metrics are not sufficient for assessing different attributes of CBSS. For working on reusability of CBSS, the most crucial task is to select the most important reusability factors and the metrics to measure these factors, but it has been found that currently no such work is present where these reusability factors and metrics can be found at one place to make this searching process fast and efficient. This paper is an attempt to summarize all these reusability sub-factors and software metrics identified from this review. This summarization is given in Tables 1 and 2 respectively and classification of software metrics for reusability sub-factors is shown in Figure 1.

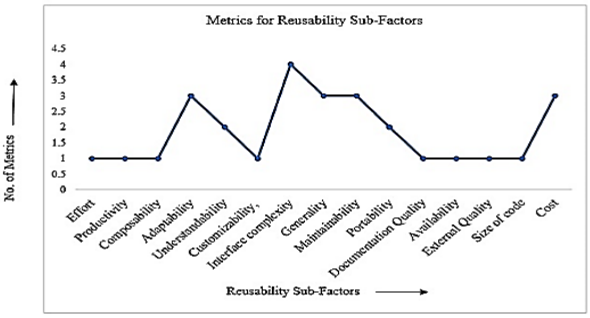

Figure 1. Classification of software metrics for reusability sub-factors.

4. Research Work Analysis

In this section, answers to research questions are given and research papers classification is also done based on different criteria.

4.1. Research Questions Answer

- *RQ*1: Chidamber and Kemerer [5] have given six metrics for Object-Oriented Software System (OOSS). Out of which five metrics can be used to assess the reusability of OOSS. From the literature review, it is identified that these metrics are not suitable for assessing the reusability of CBSS for the following reasons [25]:

a. Object oriented metrics focus on the structure of object and class, but not able to assess the component’s interfaces, coupling and cohesion.
b. Most of the object-oriented metrics based on classes but for CBSS more information is required such as interfaces among the components and interface methods.

c. Existing metrics cannot measure customizability of the classes, which must be consider while assessing reusability of CBSS.

- *RQ*2: Reusability of CBSS depends on different factors which have their own importance. These factors are proposed by different researchers and their summarization is given in Table 1.
- *RQ*3: Reusability factors, given in Table 1 are assessed using different software metrics. These metrics are summarized in Table 2, which can be aggregated to assess the overall reusability of CBSS.

4.2. Classification of Research Papers and Articles

In addition of answering the research question, research papers included in this paper are also classified according to three following criteria:

1. Research papers and articles published in different journals/ conferences: The distribution of research papers and articles included in this paper are classified based on various journals and conferences in which they are published. The details of these sources are given in Table 3 and graphical representation of these details are shown in Figure 2.

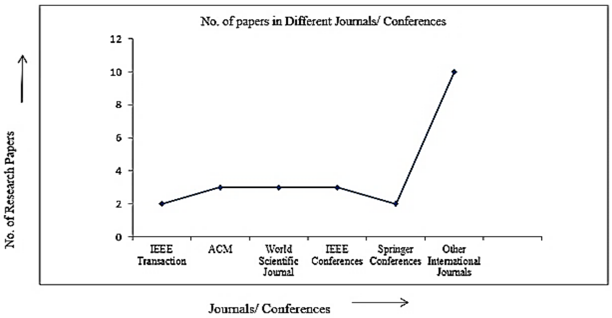

Figure 2. Classification of research papers in different journals/ articles.

Table 1. Reusability factors.

S.No	Factor Name	Definition	Relationship with Reusability (Re)	Source
1.	Generality (Ge)	Level of component's generalization	Ge α Re	[15, 16, 20, 28]
2.	Interoperability (In)	Cooperation among different components (written in different languages and running on different platforms	In α Re	[28]
3.	Understandability (Un)	capability of user to understand whether the component is suitable and how it can be used for a task and conditions of using the component.	Un α Re	[16, 20]
4.	Effort(E)	Tasks to develop a software application	E α 1/Re	[22]
5.	Cost (C)	Total cost (resource and money) to develop a software application	C α 1/Re	[11, 22]
6.	Productivity (P)	Number of software applications developed in an organization	P α Re	[22]
7.	Adaptability(A)	Acceptance of software applications among the users	A α Re	[11, 23]
8.	Composability (Com)	Composition of valid and tested components	Com α Re	[23]
9.	Interface Complexity (IC)	Used to interact between application and the components.	IC α 1/Re	[11, 23, 24]
10.	Customizability (Cu)	Ability to modify the component as per user requirements	Cu α 1/Re	[24]
11.	Portability (Po)	the ease with which a system or component can be transferred from one hardware or software environment to another	Po α Re	[20, 24]
12.	Documentation Quality (DC)	Required to understand the components specially for black box components.	DC α Re	[11, 20, 24]
13.	Maintainability (M)	the ease with which a software component can be modified to correct faults, improve performance or to adapt the changing environment	M α Re	[11, 20]
14.	Availability (Av)	how easy and fast a software component can be retrieved	Av α Re	[11]
15.	Component Quality (CQ)	Characteristic of component to describe how good the component is to fulfil the requirement and how error- and bug-free it is.	CQ α Re	[11]

Table 2. Software metrics for assessing different reusability factors.

S.No	Metrics	Reusability Factor	Source
1.	Reuse % of product	Effort	[22]
2.	Reuse Cost Avoidance	Cost	[22]
3.	Reuse Value Added	Cost	[4]
4.	Additional Development Cost	Cost	[4]
5.	Productivity Index	Productivity	[22]
6.	Composability Degree	Composability	[23]
7.	Sum of Component's adaptability itself and adaptability of context	Adaptability	[23]
8.	Existence of Meta Information	Understandability	[29]
9.	Rate of Component Observability	Understandability	[29]
10.	Rate of Component Customizability	Customizability, Adaptability	[29]
13.	No. of Interfaces, Methods and variables	Interface complexity	[26]
14.	Generality of Class (GC)	Generality	[20]
15.	Coupling Between Object (CBO)	Maintainability	[20]
16.	Number of Children (NOC)	Maintainability	[20]
17.	External Dependency	Portability	[20]
18.	Coupling	Portability, Interface Complexity	[20]
19.	Documentation	Documentation Quality	[11, 20]
20.	Number of Independent Component	Availability	[2]
21.	AD_QMOOD= -0.25 DCC + 0.25 CAM + 0.5 CIS + 0.5 DSC	Adaptability	[2]
22.	MD_QMOOD=0.25*ANA-0.5*DCC+0.5*NOH +0.5*NOP	Maintainability	[2]
23.	No. of open bugs in the issue, no. of close bugs in the issue, average rating by the user	External Quality	[2]
24.	Number of public methods	Size of code	[23]
25.	Cohesion	Interface Complexity	[11]
26.	Number of methods and parameters	Interface Complexity	[11]
27.	Domain abstraction	Generality	[28]
28.	Reuse Maturity	Generality	[28]

2. Research papers and articles focusing on reusability factors or reusability software metrics: This work includes the research papers and articles which mainly focus on reusability factors and their software metrics. These papers are classified according to their focus area (reusability factors or reusability software metrics). This summarization is given in Table 4.
3. Research papers and articles in successive years: In CBSS, research has been continued for successive years and many research are published. That why, in this paper research papers and articles are also classified as per successive years of their publication, which is given in Table 5 and graphically it is represented by Figure 3.

Table 3. Research papers published in different journals/conferences.

Journal Name/Conference	Source	Total Number of Relevant Paper Found	% Proportion
IEEE Transaction	[5, 7]	2	8.7%
ACM	[9, 14, 24]	3	13.04%
World Scientific Journal	[12, 25, 26]	3	13.04%
IEEE Conferences	[6, 9, 29]	3	13.04%
Springer Conferences	[2, 28]	2	8.7%
Other International Journals	[1, 8, 10, 11, 15, 17, 20, 23, 25, 27]	10	43.48%

Table 4. Classification of research papers as per reusability factors and reusability metrics.

Classification Criteria	Source	Total no. of included papers	% Proportion
Reusability Factors	[11, 15, 16, 20, 22, 23, 24, 28]	8	34.78%
Reusability Software Metrics	[2, 8, 11, 20, 22, 23, 28, 29]	8	34.78%

Table 5. Classification of research paper as per successive years.

Year	Source	Total no. of included papers	% Proportion
1991-1996	[4, 5, 7, 10, 22]	5	21.74%
1997-2001	[6, 8]	2	8.66%
2002-2006	[1, 9, 23, 26, 29]	5	21.74%
2007-2011	[15, 16, 17, 24, 25, 27, 28]	7	30.43%
2012-2016	[11, 20]	2	8.66%
2017-2019	[2, 12]	2	8.66%

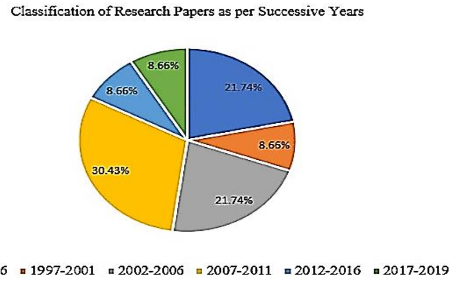

Figure 3. Classification of research papers in successive years.

5. Conclusions and Future Scope

Software companies are currently working on the development of Component Based Software System to increase the reusability and to reduce the development time. Reusability of any component can be assessed by measuring different factors of reusability. The present research work is an attempt to identify and aggregate the major reusability factors for CBSS from the existing research work and after that software metrics are identified for those factors. It has been also analyzed that reusability metrics used for object-oriented system cannot work for CBSS and the reasons for the same are also described in the paper. This paper will be useful for the software developers and researchers to select the appropriate reusability factors and metrics to continue their research in reusability of CBSS. In future, experimental work can also be done to assess the reusability of CBSS by applying latest techniques, which will allow the developers for comparing the reusability of different components and will help them to select the most reusable software component for creating a new software system.

References

[1] Allen P., "CBD Survey: The State of the Practice," *A White Paper by Cutter Consortium*, 2002.

[2] Ampatzoglou A., Bibi S., Chatzigeorgiou A., Avgeriou P., and Stamelos I., "Reusability Index: A Measure for Assessing Software Assets Reusability," *in Proceedings of International Conference on Software Reuse*, Madrid, pp. 43-58, 2018.

[3] Aris H. and Salim S., "State of Component Models Usage: Justifying the Need for a Component Model Selection Framework," *The International Arab Journal of Information Technology*, vol. 8, no. 3, pp. 310-317, 2011.

[4] Caldiera G. and Basili V., "Identifying and Qualifying Reusable Software Components," *Computer*, vol. 24, no. 2, pp. 61-70, 1991.

[5] Chidamber S. and Kemerer C., "A Metrics Suite for Object Oriented Design," *IEEE Transactions on Software Engineering*, vol. 20, no. 6, pp. 476-493, 1994.

[6] Cho E., Kim M., and Kim S., "Component Metrics to Measure Component Quality," *in Proceedings of 8th Asia-Pacific Software Engineering Conference*, Macao, pp. 419-426, 2001.

[7] Daskalantonakis M., "A Practice View of Software Measurement and Implementation Experiences within Motorola," *IEEE Transactions on Software Engineering*, vol. 18, no. 11, pp. 998-1010,1992.

[8] Dumke R. and Schmietendorf A., “Possibilities of the Description and Evaluation of Software,” *Metrics News*, vol. 5, no. 1, pp. 13-26, 2000.

[9] Gill N., “Reusability Issues in Component-Based Development,” *ACM SIGSOFT Software Engineering Notes*, vol. 28, no. 4, pp. 4, 2003.

[10] Henderson B., *Object Oriented Metrics, Measure of Complexity*, Prentice Hall, 1996.

[11] Hristov D., Hummel O., Huq M., and Janjic W., “Structuring Software Reusability Metrics for Component-Based Software Development,” *in Proceedings of 7th International Conference on Software Engineering Advances*, Portugal, pp. 421-429, 2012.

[12] Imoize A., Idowu D., and Bolaji T., “A Brief Overview of Software Reuse and Metrics,” *World Scientific News*, vol. 122, pp. 56-70, 2019.

[13] Jatain A., “Metric Based Reusability Analysis of Software Systems,” *Journal of Interdisciplinary Mathematics*, vol. 23, no. 1, pp. 107-116, 2020.

[14] Jha S. and Mishra R., “A Review on Reusability of Component Based Software Development,” *Theory and Applications*, vol. 14, no. 4, pp. 32-36, 2019.

[15] Kalaimagal S. and Srinivasan R., “Q’Facto 10-A Commercial Off-The-Shelf Component Quality Model Proposal,” *Journal of Software Engineering*, vol. 4, no. 1, pp. 1-154, 2010.

[16] Kalaimagal S. and Srinivasan R., “Q’Facto 12-An Improved Quality Model for COTS Components,” *ACM SIGSOFT Software Engineering Notes*, vol. 35, no. 2, pp. 1-4, 2010.

[17] Kitchenham B., Brereton O., Budgen D., Turner M., Bailey J., and Linkman S., “Systematic Literature Reviews In Software Engineering-A Systematic Literature Review,” *Information and Software Technology*, vol. 51, no.1, pp. 7-15, 2009.

[18] Nazir S., Shahzad S., Mahfooz S., and Nazir M., “Fuzzy Logic based Decision Support System for Component Security Evaluation,” *The International Arab Journal of Information Technology*, vol. 15, no. 2, pp. 224-231, 2018.

[19] Negi P. and Tiwari U., “Machine Learning Algorithm for Assessing Reusability in Component Based Software Development,” *EasyChair Preprint*, vol. 4142, pp. 1-8, 2020.

[20] Nyasente S., Mwangi W., and Kimani S., “A Metrics-based Framework for Measuring the Reusability of Object-Oriented Software Components,” *Journal of Information Engineering and Applications*, vol. 4, no. 4, pp. 71-84, 2014.

[21] Padhy N., Panigrahi R., and Satapathy S., “Identifying The Reusable Components From Component-Based System: Proposed Metrics And Model,” *in Proceedings of Information Systems Design and Intelligent Applications*, pp. 89-99, 2019.

[22] Poulin J., Caruso J., and Hancock D., “The Business Case for Software Reuse,” *IBM Business Journal*, vol. 32, no. 4, pp. 567-594, 1993.

[23] Rotaru O. and Dobre M., “Reusability Metrics for Software Components,” *in Proceedings of 3rd ACS/IEEE International Conference on Computer Systems and Applications*, Cairo, pp. 24, 2005.

[24] Sagar S., Nerurkar N., and Sharma A., “A Soft Computing Based Approach to Estimate Reusability of Software,” *ACM SIGSOFT Software Engineering Notes*, vol. 35, no. 4, pp. 1-5, 2010.

[25] Sharma A., Kumar R., and Grover P., “A Critical Survey of Reusability Aspects for Component-Based Systems,” *World Academy of Science, Engineering and Technology International Journal of Industrial and Manufacturing Engineering*, vol. 1, no. 9, pp. 420-424, 2007.

[26] Sharma A., Kumar R., and Grover P., “Complexity Measures for Software Components,” *WSEAS Transactions on Computers*, vol. 6, no. 7, pp. 1005-1012, 2007.

[27] Sharma A., Kumar R., and Grover P., “Managing Component-Based Systems with Reusable Components,” *International Journal of Computer Science and Security*, vol. 1, no. 2, pp. 52-57, 2007.

[28] Upadhyay N., Despande B., and Agrawal V., “Towards A Software Component Quality Model,” *in Proceedings of International Conference on Computer Science and Information Technology*, Bangalore, pp. 398-412, 2011.

[29] Washizaki H., Yamamoto H., and Fukazawa Y., “A Metrics Suite for Measuring Reusability of Software Components,” *in Proceedings of 5th International Workshop on Enterprise Networking and Computing in Healthcare Industry*, Sydney, pp. 211-223, 2004.

Jyoti Agarwal obtained her Ph.D. Computer Science from Amity University, Noida. She is gold medalist in M.Tech(CSE) and completed her B.Tech in 2010 from UPTU. She is currently working as Assitanet Professor in AMity University Noida since 2011 and also has 6 months of teaching experinece in Amity University Tashkent. She has published many research papers in reputed journals and conferences. She is a member of International Association of Engineers (IAENG). Her current research interests include Software Engineering, Data Mining and Machine learning.

Manoj Kumar obtained his Ph.D. Computer Science from The Northcap University, Gurugram. He obtained M. Sc. (Information Security and Forensics) degree from ITB, Dublin. Mr. Kumar has 9.5+ years of experience in research and academics. He published over 43 publications in reputed journals and conferences. Presently, Mr. Kumar is working on the post of assistant professor (SG), in University of Petroleum and Energy Studies, Dehradun. He is a member of various professional bodies and reviewed for many reputed international journals. His current research interests include Digital Forensics, Information Security, Image Processing, Networks, Internet of Things (IoT) and Blockchain technology