A Survey On Analysis And Design Of Component Based Metrices

Isha Rani[1],Pinki Tawnar[2]
Department of Computer Science and Engineering,JMIT
Radaur,India
[1] trishamandhan92@gmail.com
[2]Pinkitanwar@jmit.ac.in

***Abstract*— CBSE is a process that emphasize the plan and building of computer based systems using reusable software components. The ground of software systems has become increasingly complex and performance precise. To produce cost-effective systems, organizations often make use of component-based technologies as an substitute to rising the whole system from scratch. Build up representative dataset of projects(object-oriented). Identify parameters which impact the changeability & reusability of software components. Develop a framework and to measure the changeability, reusability metrics**

Keywords- CBSE, System level metrics,Component level metrics,Clustring,Software Components.

I. INTRODUCTION

INTRODUCTION TO COMPONENT BASED SOETWARE ENGINEERING(CBSE)

CBSE is a process that emphasize the plan and building of computer based systems using reusable software components. The ground of software systems has become increasingly complex and performance precise. To produce cost-effective systems, organizations often make use of component-based technologies as an substitute to rising the whole system from scratch. The purpose behind using component-based technologies is to decrease the charge connected with the development. Nevertheless, this field later evolved into a more serious area to reduce reliance on the current market and to match rapidly rising consumer requirements. Currently, the custom of component-based technologies is additional frequently compulsory towards curtailing the development expenses.[1]

SOFTWARE COMPONENT

Software Component is a wrap up of software implementations which provides well clear interfaces. Softwar components can be a chunk of code, unit, function or class, scheme or software itself and when these components get incorporated they form an whole application [2].

COMPONENT CHARACTERISTICS

1. **Component Suitability**

The component suitability can be assessed by two attributes that are the Required Functionality (RF) and Extra Functionality (EF). amid both of them the more the value of RF, extra will be the component's suitability. This earnings that the component is provided with additional requirements for the system. However the worth for EF has to be slighter for superior suitability of the component.

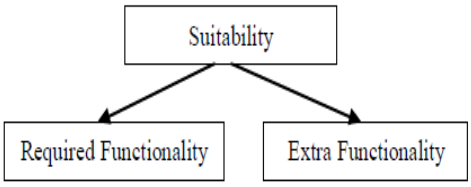

FIGURE 1: COMPONENT SUITABILITY TREE [2]

2. **Component Accuracy**

The exactness of the component indicates scope of accuracy to be acchived in satisfying the supplies and anticipated functionality of the component.

3. **Component Complexity**

The complexity of a software module is the assess the internals of the components to price the quality characters supported by the components. The component internals includes the classes, interfaces as well as their provision

The component complexity is the character, which have an impact directly on the output of the components. The complexity of the component is unfair by the necessities of the components. The more the quality character probable from the component, added will be the complexity of it.

4. Component Usability

It is the capability of the software component to be learned, used, understood and is also attractive to the user in different context of use. Usability is a characteristic that will be looked upon for assessment by the producers as well as consumers.

5. Component Maintainability

Maintainability defines the extent of modification of a software component and that too after its deployment. Modifications include corrections, improvements etc to the software components due to the changes in requirement or in environment. As far as the modifications are concerned, it is usually carried out by the components producers and they may use the traditional approaches of its maintenance.

6. Component Reusability

Reusability of the component is the extent to which the component can be reused in the other as well as same component based applications. It can either be the reuse that can be achieved for developing a component or the level of reuse from the component for the component based applications.

7. Component Performance

Performance of the component is measured in terms of its response time as well as throughput in performing its functionality. The Performance of the component can be measured through two attributes namely the Response Time and the Throughput of the component. Both of them can be measured only after deploying the components. Hence they can be tabulated for different context of use of the component and can be used as a feedback for easy going tuning of the performance in successive versions of the component .

CBSE Metrics

The CBSE metrices can be defined at two levels these two levels can be specified as below.

- System Level Metrices
- Component Level Metrices

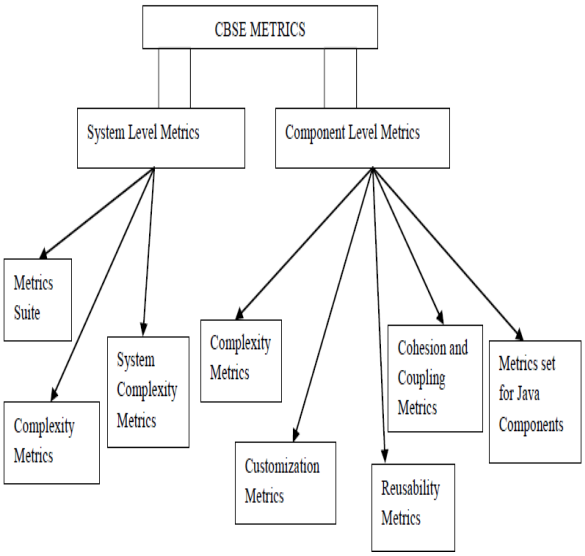

Figure 2. CBSE Metrices[3]

System Level Metrics

Metric Suite:-Metrics for component based systems are in three categories: management, quality and requirements. Management associated metrics contain time to market, cost of product, Software source deployment and maturity of software progress atmosphere. Requirement related metrics include:,requirements

permanence and requirement conformance. Quality related metrics include: difficulty of interface, adaptableness, end-to-end test coverage, incorporation test coverage, cumulative number of detected faults, customer satisfaction level and reliability.

Complexity Metric:-In a component based system, complexity outcome from dependencies between the components of the system. This metric uses inter-component reliance information to establish the complexity of a component based system. classification of the metric depend on the idea of Component Dependency Graph (CDG) whose vertices characterize components, and edges represent the dependencies between components. The CDG is represented in the figure of a matrix, with cells of the matrix containing a value 1 if dependency exist in the equivalent component pair and or else a value 0

System Complexity Metric: - some metrics are defined to calculate a component based system mostly concentrating on its structural complexity. most important attributes that verify complexity of structure of a component based system are recognized as:, connectors, mechanism, composition tree and interfaces.

Component Level Metrics

Component level metrics help out to calculating the component superiority in terms of its complexity, reusability and customizability.

Component Complexity Metric:-It can be Component Static Complexity (CSC), Component Plain Complexity (CPC), Component Cyclomatic Complexity (CCC) and Component Dynamic Complexity (CDC). The CSC metric determines the complexity of inner structure of a component. It is the weighted amount of various types of relationships in a component. The CPC metric is the sum of essentials of the component (abstract classes, classes and interface), further complication of all classes, and additional complexity of all methods of the classes. As compared to further metrics which are existing at design stage, the CCC metric is available after implementation. It resemble to the CPC metric, only with a variation that it uses McCabe's complexity metric to verify complexity of method of a class The CDC metric focus on the complexity of message passing occurring internally in a component.

Component Customization Metric:-It determines the variability of the method in a component's boundary. Metric is the proportion of extent of methods for customization to the total number of methods announced in all the interfaces of the component.

Component Reusability Metric: - It calculates the reusability of a component at design stage of the component improvement process. The metric, component reusability (CR) can be calculated as proportion of sum of interface methods given that common functionality in a domain to the total number of interface methods presented in a component. The idea to determine these metrics have been taken from the banking domain. The metrics also present the indication of some properties of a software component such as maintainability, understandability, and reusability.

Metrics Set For JAVA Components:-A position of metrics to calculate the reusability property of black box components which are accommodating to Java Beans Component model. They describe following five metrics for calculating different values that contribute to the reusability of a software component: continuation of meta-information, component observability, external dependency and customizability of components.

Component Cohesion and Coupling Metrics - The dynamic dependency associations among classes indicates the high cohesion between components. The cohesion metric are interested in consideration the structural relationships alongside with the types of method involving classes of an object oriented component. Coupling between classes Cn, Cm, denoted by CC (Cn, Cm), is define as the weighted sum of dissimilar types of method called among both the modules of classes. Component coupling is also defined as the sum of coupling with all pairs of classes of the component.

Contextual Reusability metric – The metric calculates the component reusability is dissimilar from other metric in this category. The scheme is that in addition to component reusability and internal attributes.it also depends on the context in which it is reuse. So their component reusability metric depend on the component's compliance to diverse elements of the architecture of an application in which it is to be included. Metrics belonging to this category are: Component observance Metric and Architecture observance Metric.[3]

CLUSTERING - Cluster investigation is the process of the assignment of a position of observations into the subsets/ clusters. Using dissimilar clustering techniques , a variety of different assumptions on the structure of

the data are completed. Defined by some parallel metrices are measured by internal compactness i.e the similarity among members of the matching cluster and separation among other clusters. It is a main task of data mining, a common technique for statistically analyzing the data is generally used in several fields, including pattern recognition, machine, bioinformatics & image analysis

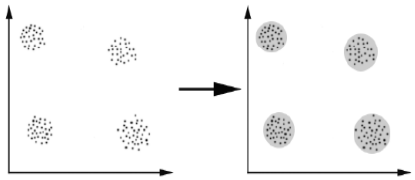

FIGURE 3: CLUSTER FORMATION[2]

II. LITERATURE SURVEY

V. Lakshmi Narasimhan et al(2009)[4] Component-Based Software Engineering (CBSE) has revealed important prospects in quick productionof huge software systems with improved quality, and emphasis on breakdown of the engineered systems interested in functional or logical components with well-defined interfaces worn for communication across the mechanism. In this paper, a sequence of metrics planned by a range of researchers have been analyzed, evaluated and benchmarked with several big-scale publicly existing software systems. A systematic analysis of the standards for different metrics has been accepted out and a number of key inferences have been drained from them. A numeral of useful conclusions have been strained from a variety of metrics evaluations, which contain inferences on reusability,complexity,modularity,testing and stability of the original components. The inferences are argued to be helpful for CBSE-based software development, addition and protection.

P. Edith Linda et al(2011)[5] The main plan of this paper is to add the diverse object oriented metric apparatus and create them accessible as a single add-on. The first fraction of this document analyzed five dissimilar tackle and they are migrated interested in one to construct use of those gear in efficient style.

Prakriti Trivedi et al (2012)[6]Today the majority of the applications residential using a number of codes, existing libraries, open sources etc. As a strategy is accessed in course, it is represented as the software module Such as in .net ActiveX controls and java beans are the software mechanism. These components are complete to use programming rules or controls that excel the system growth. A component based software organization defines the perception of software reusability. While by means of these mechanism the main question occur is whether to use such components is helpful or not. In this planned work we are tiresome to present the reply for the similar question. In this occupation we are presenting a position of software matrix that will verify the interconnection among the software element and the application. How well-built this relative defines the software value after using this software part. The generally metrics will revisit the final product in terms of the unlimited of the part with application.

Majdi Abdellatief et al (2013)[7]A component-based software system (CBSS) is a software organization that is urbanized by integrating mechanism that have been deployed separately. In the final few years, numerous researchers have planned metrics to estimate CBSS attributes. though, the useful use of these metrics can be not easy. For instance, several of the metrics have concepts that also overlie or are not fine defined, which could hold back their execution. The plan of this study is to recognize categorize and examine existing study in component-based metrics, center on approaches and essentials that are worn to appraise the quality of CBSS and its mechanism from a module consumer's point of vision. This paper presents a organized mapping study of numerous metrics that were planned to calculate the excellence of CBSS and its components. Author found 17 proposals that could be functional to appraise CBSSs, while 14 proposals could be useful to evaluate character components in separation. diverse elements of the software components that were calculated are reviewed and discussed. Only a a small number of the projected metrics are completely distinct. The excellence estimation of the principal studies detected many restrictions and suggested strategy for potential for humanizing and growing the recognition of metrics. However, it remainder a confront to illustrate and assess a CBSS and its components quantitatively. For this motivation, much attempt must be ended to accomplish a better evaluation move on the way to in the future.

Anshul Kalia et al (2014)[8]The reusable software components can be obvious in numerous ways. The reusable software components acquire a distinct functionality that does not affect the functionality of other components. It has also been specified exactly that for what the component reuse stands for and for what the component use again does not stands for. It is necessary to differentiate the components for enhanced reuse.

Characterization describes the features and uniqueness of components. Distinct components show different characteristics in diverse domains of their usage and in different working environments. The components can be classify on the basis of features it have, that facilitates with the improved usage, better retrieval, better perceptive and better classification. Through component classification one gets the guarantee of selecting right component and it suggests different ways in which a component can be reused. The paper explain the necessitate of characterization which ultimately is reflected from the more than stated fact. It also provide with the criteria to illustrate the reusable components. The criterion is laid down while trust in mind the common specifications and official specifications. These specifications in one way decide the feasibility of a reusable component at the original level. The general and formal specifications show the internal and external characteristics which play a important role in the selection, identification, adoption and execution of components in a particular application progress. Specifications will tell the right character of a reusable component that helps to outline the essential components. It also discusses the manipulation of characterization that it puts on the recycle of reusable components.

ZHANG Min et al (2015)[9] K-means in the field of clustering examination algorithms is a kind of further traditional algorithm. It exist many shortcomings. For example, K importance is easily affected by manmade personal factors, and the algorithm is simple to fall into a local optimal result, and the clustering result is not constant, etc; And K-means++ algorithm as the classic enhanced algorithm of K-means algorithm, but there is still a experience of unstable cluster center. This paper is a kind of development aimed at the shortcoming of K-means++ algorithm, which introduce the concept of the variation in probability and mathematical statistics. Variance reflect the degree of thickness between samples and other samples. In the K-means++ algorithm when you choose the first original clustering center, you need to select least variance of sample points, which is in the location of the largest sample density, then you pick the next cluster centers based on the weight method of D2 which is describe in the K-means++ algorithm. Experimental outcome show the correctness is higher and stability is better.

Shweta Bhambri et al (2016)[2] Reusability of the component denote the extent to which the component can be reuse in the other as well as similar component based applications and the difficulty of a software component is about evaluate the internals of the components to verify the value characters support by the components. When a car is manufactured, thousands of components are developed to get the final whole car. Most of the components of the cars are developed in such a way that these components are interchangeable and can be used in any another car if needed and if one component is not functioning properly it can be interchanged with other component by plugging in and out with other components, so that changeability of the components is increased and maintenance can be made easier. There is an urgent need to understand how these software components can be implemented as plug and play devices and reusability as well as the changeability of software components can be understood in some framework. Therefore, in this research work we are solving this type of issue by building a framework which helps to measure degree of changeability & reusability by using clustering methods, which would be best suitable for our problem and will be better from previous research work.

III. SUMMARY OF THE TECHNOLOGIES AND DIFFICULTIES FACED

THE GIVEN TABLE CONTAINS SOME OF TECHNIQUES USED ALONG WITH THEIR AUTHORS AND METHOD GIVEN BY THEM.

AUTHOR	METHOD/APPROCH	YEAR	FINDINGS
Shweta Bhambri	Estimation of Software Reusability Based on Clustering	2016	proposed approach calculates the reusability and complexity of the components (samples
Chander Diwaker	Various metrics proposed to measure the different attributes of a component like functionality, interactivity, complexity, reusability etc.	2014	Component characterization is necessary for better understanding of architecture, better usage, better retrieval, better cataloging along with improvement in software reusability.
Majdi Abdellatief	analyzed existing	2013	An interesting area

	research in component-based metrics, focusing on approaches and elements that are used to evaluate the quality of CBSS and its components from a component consumer's point of view		for further research involves revising the existing definition of CBSS metrics for better precision in measurement.
P.Edith Linda	integrated the different object oriented metric tools and make them available as a single add-on.	2011	a number of algorithms was presented along with a comparative study of a few significant ones based on their performance and memory usage

IV. CONCLUSION

In this paper we have looked at various metrices and clustering techniques.we have reviewewd and provided a detailed explanation of the metrics**.** Estimation of Software Reusability Based on Clustering proposed approach calculates the reusability and complexity of the components (samples). Various metrics measure the different attributes of a component like functionality, interactivity, complexity, reusability etc. Component characterization is necessary for better understanding of architecture, better usage, better retrieval, better cataloging along with improvement in software reusability. Existing research in component-based metrics, focusing on approaches and elements that are used to evaluate the quality of CBSS and its components from a component consumer's point of view An interesting area for further research involves revising the existing definition of CBSS metrics for better precision in measurement integrated the different object oriented metric tools and make them available as a single add-on. Various algorithms was presented along with a comparative study of a few significant ones based on their performance and memory usage.

REFERENCES

[1] Adnan Khan, Khalid Khan, Muhammad Amir and M. N. A. Khan " **A Component-Based Framework for Software Reusability"** International Journal of Software Engineering and Its Applications Vol. 8, No. 10 (2014), pp. 13-24 http://dx.doi.org/10.14257/ijseia.2014.8.10.02

[2] Shweta Bhambri1, Sheetal Chhabra2" Estimation of Software Reusability Based on Clustering" DOI 10.4010/2016.1682 ISSN 2321 3361 © 2016 IJESC

[3] Chander Diwaker, Sonam Rani, Pradeep Tomar '**Metrics Used In Component Based Software Engineering' IJITKM**Special Issue (ICFTEM-2014) May 2014 pp. 46-50 (ISSN 0973-4414)

[4] V. Lakshmi Narasimhan " Evaluation of a Suite of Metrics for Component Based Software Engineering (CBSE)" Issues in Informing Science and Information Technology Volume 6, 2009.

[5] P. Edith Linda "Metrics for Component Based Measurement Tools" International Journal of Scientific & Engineering Research Volume 2, Issue 5, May-2011 1ISSN 2229-5518

[6] Prakriti Trivedi **"** Software Metrics to Estimate Software Quality using Software Component Reusability" IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 2, No 2, March 2012 ISSN (Online): 1694-0814 www.IJCSI.org.

[7] Majdi Abdellatief "A mapping study to investigate component-based software system metrics" The Journal of Systems and Software 86 (2013) 587– 603

[8] **Anshul Kalia "** Characterisation of reusable software components for better reuse IJRET: International Journal of Research in Engineering and Technology eISSN: 2319-n 1163pISSN: 2321-7308

[9] ZHANG Min " Improved research to k-means initial cluster centers" 2015 Ninth International Conference on Frontier of Computer Science and Technology

[10] Hasan Kahtan, Nordin Abu Bakar, Rosmawati Nordin, "Reviewing the Challenges of Security Features in Component Based Software Development Models",IEEE Symposium,2012

[11] Chen Xingshu,et al.,"K-means initial clustering center optimal algorithm based on feature relevance", Journal of Sichuan University (Engineering Science Edition), 2015 (01): P. 13-19.

[12] DivyaChaudhary, Prof. Rajender Singh Chillar, "Component Base Software Engineering Systems: Process and Metrics", International Journal of Advanced Research in Computer Science and Software Engineering, July 2013, Vol. 3, Issue 7, pp. 91-95.

[13] Jianguo Chen, Hui Wang, Yongxia Zhou, Stefan D. Bruda "Complexity K. Metrics for Component-based Software Systems", International Journal of Digital Content Technology and its Applications, Volume 5, 2011.

[14] T. U. Rehman, M. N. A. Khan and N. Riaz, "Analysis of Requirement Engineering Processes", Tools/Techniques and Methodologies. International Journal of Information Technology & Computer Science, vol. 5, no. 3, (2013).

[15] Gurdev Singh, Dilbag Singh, Vikram Singh, "A Study of Software Metrics", International Journal of Computational Engineering & Management, Jan 2011, Vol. 11, pp. 22-27.