International Journal of Enhanced Research in Management & Computer Applications
ISSN: 2319-7471, Vol. 5 Issue 6, June-2016, Impact Factor: 1.544

A Feature Adaptive Measures of Reusability for Real Software Systems

Sonal[1], Gagandeep[2]

[1]Student, M.Tech. (CSE), Prannath Parnami Institute of Management and Technology, Hisar, Haryana
[2]Asstt. Prof, CSE Dept. Prannath Parnami Institute of Management and Technology, Hisar, Haryana

ABSTRACT

The quality analysis on a software system can be defined under different aspects. One such evaluation measure is software reusability. In this present work, a feature adaptive analysis on reusability is provided software systems. The model accepted the real software project as input and performs the extraction of components of different types. These components are processed under high level and low level measures. The algorithmic model of work is presented. The experimentation is provided on some real project and evaluates the complexity in reusability of component based project.

Keywords: Component Model, Evaluation, CBSE, Software Quality

I. INTRODUCTION

A component based software system is the more organized and structured architecture for software system generation. The model is defined for generating the components along with integration specification of each object as well as its integration to the software projects. The software system is defined in the form of integrated objects, modules or the components which are developed either from the start or taken from some other software system. The component specific evaluation can be performed for individual estimation so that the application driven analysis will be performed. The component generation, modeling, integration and reusability are the key processes defined with component driven programming. The software system evaluation can be performed with specification of these software components. The component based engineering not only improves the effectiveness of software system but also provides the support with multiple associated features including the scalability, reusability, robustness and the reliability. The processing of this architecture is defined in an organized and structural form. The evaluation is here provided under different feature estimation. The cost, quality and the efficiency are the measure estimated at each level of software system development. To design such a system, the first work is to define the software component. A software component can be presented as an entity with associated properties, methods and events. The component can be defined as a module or class or the control depending on the application. Some of the common categories of components are shown in figure 1
.

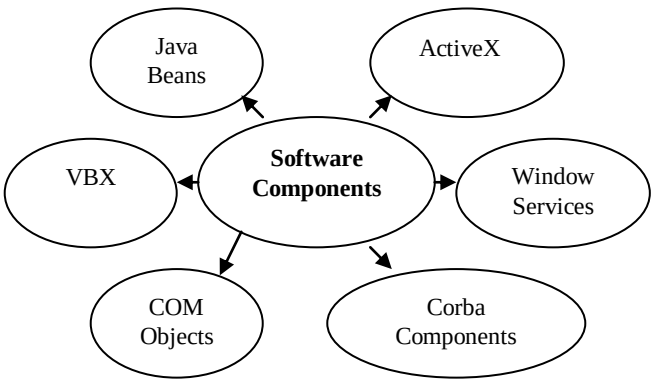

Figure 1: Different Forms of Software Components

Here figure 1 is showing the different forms of software components depending on the language or the environment. These components are defined in different languages such as VB, .net, java etc. These components are either code based or control based.

A) Software Reusability

Software components can be used in other software systems or products to provide an extension to the system. These existing software components can be used to improve the reliability, quality and productivity of the software system. The cost and time specific evaluation is provided. The reuse process analysis and the relational observation can be performed to obtain the structural reuse. The use of ready components or projects in new software system increases the quality and reliability of the product. The new product needs not to analyze completely but only the selected updated segments can be analyzed. The component level reusability will be achieved by providing the component usage and to provide the reduce the development and integration risk.

B) Measures

As the software system is defined in the form of various integrated component defined with relative intercommunication. The characterization on these components is required to achieve the component usage and the quality assessment. Different measures can be characterizing the features in numerical form. These features includes the performance level features, reliability level feature, risk weighted values etc. The product level, process level, integration level measures can be defined to perform such evaluation. The cost estimation and evaluation is required to represent the measure of the component. Each of the associated integration, communication and interconnection is also valued by these measures.

In this paper, a quality estimation of reusability process is provided. The method is applied on code specific projects and applied to observe the quality under different measures. In section I, the characterization of component based software system is defined along with the reusability features specification and defining the different measures of software component evaluation. In section II, the work defined by earlier researchers is discussed. In section III, the proposed research methodology is defined. In section IV, the results related to work are provided. In section V, the conclusion of work is presented.

II. RELATED WORK

Software quality analysis, evaluation and estimation can be performed under different features and aspects. Many researchers have already worked on component based software engineering and the quality analysis. The reusability is one such vector respective to which the software quality is analyzed. In this section, the work of earlier researcher for component based software system and the estimation of software quality for usability is presented. Author[1] has defined a work on failure prediction of online component based software products. Author identified many of existing problems for larger project by observing the architecture, fault prediction and system driven observation. The compositional models were provided to model these features separately and to generate the feature transition modeling. The failure prediction and defining the software component separately was also provided by the author. Author[2] defined a verification integrated model with integrated heuristic technique to provide the quality feature analysis. Author obtained the state explosion problem with formal verification for large software system.

The interleaved behavior of software products with asynchronously actions was observed to generate the state specific properties. The deadlock detection, reachability measure and live lock monitoring was observed as the integrated component. Author[3] has provided some of the measure to analyze and verify the reusability for a software system. Author identified some of the factors in structure specification and generation to analyze the quality and the aspect driven observations. The judgment metrics was constructed to improve the reusability aspect. The characterization was provided by the author improve the reusability in the state flow analysis. Author[4] has defined a quality model for service level estimation under reusability concern. The estimation was provided to analyze the software services and to derive the system parameters. The partial constraint matching and the adaptive integration were also provided to analyze the behavior of existing software product.

Author[5] has defined a component level evaluation using black box technique. The component relative features including the adaptability, portability and understandability are observed to evaluate the software system quality. Various evaluation models are applied under reusability concern to generate the more adaptive results. Author[6] has provided a work on feature level prioritization and software system improvement under reusability concern. An evaluation with component adoption was provided with different complexities. The development level, organization level and project level estimation was taken to generate the attributes and to apply the measure. The space, time and cost are evaluated for each measure to identify the validly of the software system. Author[7] has defined a work to analyze the software system component under different approaches to evaluate the software product and to provide the estimation relative to associated measures.

The complexity based comprehensive measures are applied to observe the structure and to generate the statistical notifications relative to the component quality. Some of the recommender measures are applied to observe the quality

of software products. Author[8] has defined a fuzzy logic based method to estimate the software product with specification of different membership functions. Author analyzed the membership under complexity, modularity, interface complexity and flexibility parameters. The fuzzy set formation has categorized the features and provided the more effective component integration and analysis so that the adequate accuracy is obtained. Author[9] has defined the coupling and cohesion based evaluation on software system under reusability measure to estimate the risk. Author applied the cohesion and coupling analysis for java projects. Finally, the regression modeling is applied to generate the domain driven results. The analysis was provided to measure the risk factors and to provide the evaluation to improve the testing aspects. The effort driven estimation and design improvement was provided by the author. Author[10] has defined an analytical study to identify the usage of reusability for web projects.

The automated script was defined via macros to identify the reusability estimation. The approach is able to provide the execution of complex tasks with specification of automated process execution. The keyword query based execution is provided to obtain the task usage. Author[11] has provided the aspect driven assessment for object oriented program to measure the inheritance for reusability. The method used the UML diagram based process flow along with behavior specific analysis to generate the axiom support for the software system under different concerns. The inheritance depth class estimation and the internal property assessment were provided by the author. Author provided method specific process flow estimation by generating the inheritance structure integration. The metric level evaluation was provided to improve the quality of reusability. Author[12] has defined a language feature based estimation to improve the quality of reusability for any software system.

The software system architecture was here analyzed under reusability concern. The component integration based framework analysis was provided to generate the dependency links. The context driven estimation and software system customization was provided to generate a new software system. The interface level, class level and module level composition was discussed by the author. Author[13] has identified the concept of component as a service to evaluate the reusability in a software system. The comprehensive method defined here has provided the metric specific evaluation and attribute formulation to provide the real time assessment.

The quantitative measures are defined to generate these metrics and to identify the quality of reusability. Author[14] has provided a reusability measuring method for object learning for real time software system. The work is here defined for ELearning project to generate the inheritance tree. The method integration and the learned component evaluation were provided to identify the degree of membership and reusability. A depth and couple driven object mapping was provided to analyze the complexities of software system. Author[15] has defined a work on different features of reusability respective to quality estimation. The component level productivity estimation and relative probabilistic measures are observed to perform the reusability prediction. The component specific evaluation was provided by the author.

III. RESEARCH METHODOLOGY

CBSE defined an organized software development architecture with specification of elements, forms and the relations. To define the software product, it is required to generate the different architectural diagrams including the element architecture specification, data element specification and connecting elements. The constraint specification architectural control is provided for the component interaction analysis. One of the major advantage of this architecture is to provide the easy requirement specific search, partial extraction and software splition. Based on these all features, some existing software project can be considered to generate a new software product. This concept of generation of new software system from existing is called software reusability.

Software reusability can be performed easily at component level by adding some new components or removing the unrequired components from the system. As a new component is included in the software system, it is also required to generate the relative connectivities with existing components so that the complete software system will be reformed. In same way, as some component is removed, it is required to observe without the particular component the system is performed perfectly. To provide the effective usage of software components and to define a new software system from existing software system, it is required to analyze the initial software product under different evaluation measures. One such method for software system evaluation is defined in this paper.

This paper is basically focused to perform a pre-analysis on software system to identify the possibility of reusability under cost adaptive estimation. Sometimes, the internal software

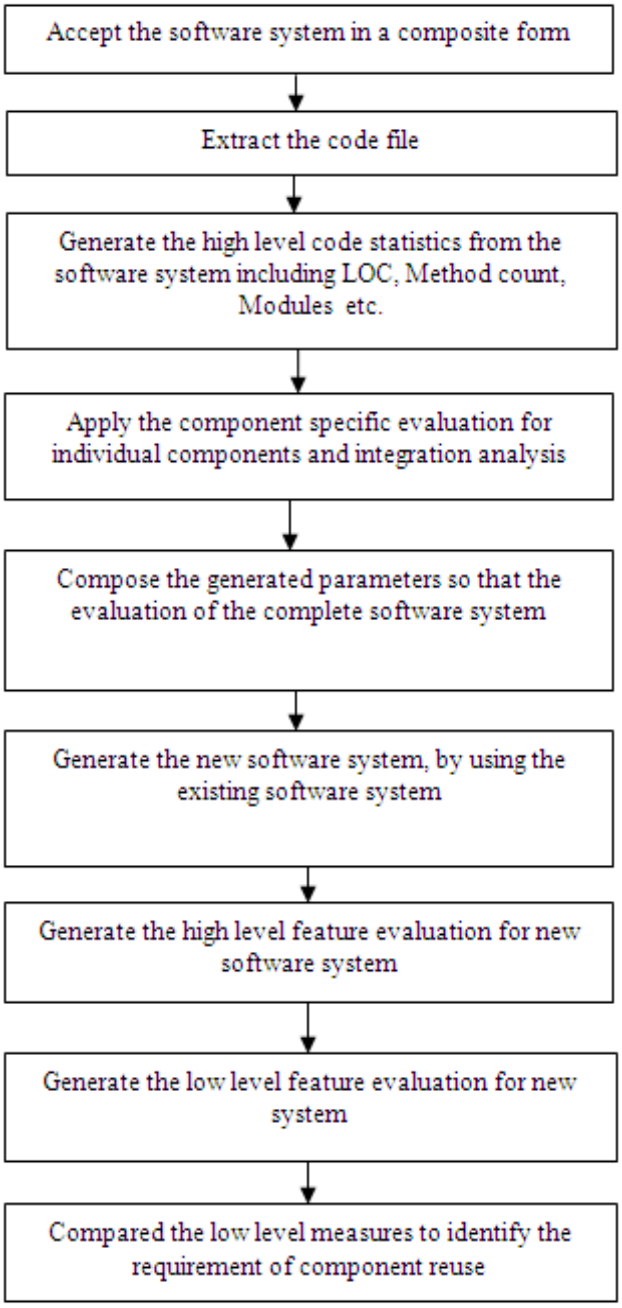

Figure 2 : Proposed Model

system architecture can be complex enough, so that it comes difficult to penetrate it and to remove the existing components or to add a new component to it. In this paper, the evaluation is here provided under different vectors so that the software system estimation will be obtained. The estimation is here applied on existing software system as well as new composed software system. The provided work is applied on real time software system under different evaluation measures. Here the process of evaluation of new software system composition is shown in figure 2.

Here figure 2 is showing the proposed work model to evaluate the effectiveness of component reuse. The proposed model has accepted the complete software project as input and defined a series of processes to extract the components from it. The evaluation is here provided in terms of components and relative feature specification. This high level analysis represented the system as the structural formulation. After generating the outer structure, the low level analysis was applied based on the component itself as well as component communication. This evaluation is performed at initial level and after composing the new software system. The parameter specific estimation is performed to identify the requirement of software reuse. The work is implemented on sample real time project. The implementation results obtained from the work is shown in next section.

IV. RESSULTS

In this present work, a effective feature adaptive method is defined to provide the analysis of reuse quality for a software system. The presented work is implemented in .net environment to accept the software project and to perform the low level component evaluation. The work is applied on a university project with some modules relative to student, exam etc. The analysis of the work is here done under low level evaluation applied after and before software reuse. The generated evaluation results are shown here in figure 3 and figure 4

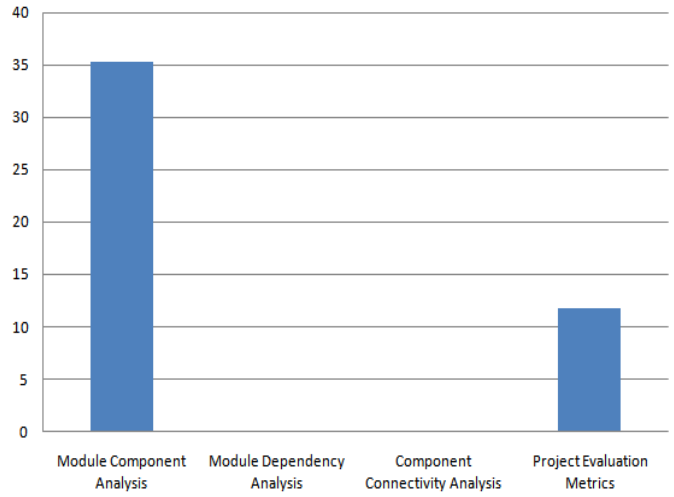

Figure 3 : Before Reuse Evaluation

Here figure 3 is showing the evaluation results of existing software system before applying the software reuse. According to this presented method, the evaluation shows that the component connectivity is lesser. It concludes that the software can be reused easily without much complications or the complexity. The results after the reuse are shown here in figure 3.

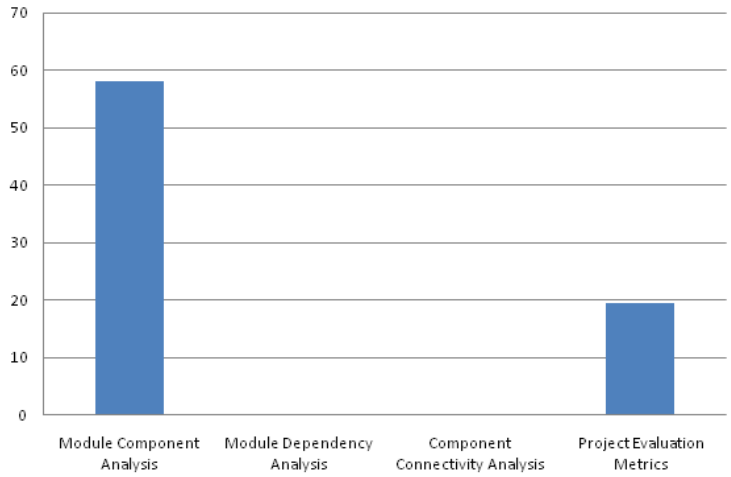

Figure 4 : After Reuse Evaluation

Here figure 4 is showing the evaluation results of new composed software system after applying the software reuse. According to this presented method, the evaluation shows that the component complexity is high and the connectivity is low. It concludes that the software that the software system will not disrupt the software system will be integrated easily in the existing software.

CONCLUSION

In this paper, an effective feature driven method is defined for estimation of quality of reuse for component based software system. The paper has defined the significance of component based software system and its evaluation. The work is applied on real time software directly to generate the high level and low level features. The evaluation shows that the method is able to provide the decision about the usage of reusability of any real time software product.

REFERENCES

[1]. T. Pitakrat, A. v. Hoorn and L. Grunske, "Increasing Dependability of Component-Based Software Systems by Online Failure Prediction (Short Paper)," Dependable Computing Conference (EDCC), 2014 Tenth European, Newcastle, 2014, pp. 66-69.
[2]. J. J. P. Tsai and E. Y. T. Juan, "Model and heuristic technique for efficient verification of component-based software systems," Cognitive Informatics, 2002. Proceedings. First IEEE International Conference on, 2002, pp. 59-68.
[3]. Zhang Juan, L. Cai, W. Tong, Yuan Song and Li Ying, "Test Case Reusability Metrics Model," Computer Technology and Development (ICCTD), 2010 2nd International Conference on, Cairo, 2010, pp. 294-298.
[4]. S. W. Choi and S. D. Kim, "A Quality Model for Evaluating Reusability of Services in SOA," 2008 10th IEEE Conference on E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services, Washington, DC, 2008, pp. 293-298.
[5]. H. Washizaki, H. Yamamoto and Y. Fukazawa, "A metrics suite for measuring reusability of software components," Software Metrics Symposium, 2003. Proceedings. Ninth International, 2003, pp. 211-223.
[6]. M. I. A. Efat, M. S. Siddik, M. Shoyaib and S. M. Khaled, "Feature prioritization for analyzing and enhancing software reusability," Informatics, Electronics & Vision (ICIEV), 2014 International Conference on, Dhaka, 2014, pp. 1-5.
[7]. J. S. Poulin, "Measuring software reusability," Software Reuse: Advances in Software Reusability, 1994. Proceedings., Third International Conference on, Rio de Janeiro, 1994, pp. 126-138.
[8]. C. Singh, A. Pratap and A. Singhal, "An estimation of software reusability using fuzzy logic technique," Signal Propagation and Computer Technology (ICSPCT), 2014 International Conference on, Ajmer, 2014, pp. 250-256.
[9]. M. Iyapparaja and S. Sureshkumar, "Coupling and cohesion metrics in Java for adaptive reusability risk reduction," Sustainable Energy and Intelligent Systems (SEISCON 2012), IET Chennai 3rd International on, Tiruchengode, 2012, pp. 1-6.
[10]. J. Admire, A. A. Zawwad, A. Almorebah, S. Karve and C. Scaffidi, "Code you can use: Searching for web automation scripts based on reusability," 2014 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), Melbourne, VIC, 2014, pp. 81-88.
[11]. G. Makkar, J. K. Chhabra and R. K. Challa, "Object oriented inheritance metric-reusability perspective," Computing, Electronics and Electrical Technologies (ICCEET), 2012 International Conference on, Kumaracoil, 2012, pp. 852-859.
[12]. R. L. Biddle and E. D. Tempero, "Understanding the impact of language features on reusability," Software Reuse, 1996., Proceedings Fourth International Conference on, Orlando, FL, USA, 1996, pp. 52-61.
[13]. H. J. La, J. S. Her and S. D. Kim, "Framework for evaluating reusability of Component-as-a-Service (CaaS)," 2013 5th International Workshop on Principles of Engineering Service-Oriented Systems (PESOS), San Francisco, CA, 2013, pp. 41-44.
[14]. S. F. M. Noor, N. Yusof and S. Z. M. Hashim, "A Metrics Suite for Measuring Reusability of Learning Objects," 2009 Ninth International Conference on Intelligent Systems Design and Applications, Pisa, 2009, pp. 961-963
[15]. P. S. Sandhu, Aashima, P. Kakkar and S. Sharma, "A survey on Software Reusability," Mechanical and Electrical Technology (ICMET), 2010 2nd International Conference on, Singapore, 2010, pp. 769-773.