International Journal of Innovative Research in Science and Engineering
Vol. No.3, Issue 01, January 2017
www.ijirse.com
ISSN (O) 2454-9665
ISSN (P) 2454-0663

A PARAMETER SPECIFIC CLUSTERING METHOD TO ANALYZE THE QUALITY OF SOFTWARE SYSTEM

Kavita[1], Preeti Sharma[2]

[1]*Research Scholar, Computer Science and Application Department, MDU, Rohtak*

[2]*Assistant Professor, Computer Science and Application Department, MDU, Rohtak*

ABSTRACT

A real time software system is composed of multiple integrated modules. The quality of the software system depends on failure rate or impact. In this paper, a cluster based parametric method is provided to recognize the high and low quality modules. The parameter considered in this work includes the failure type, failure impact, module impact and the fault frequency. Based on these vectors, the weighted formula is defined to quantize the feature of each module. After representing the modules in terms of single quantitative aspect, the KMeans clustering is applied to recognize the high and low quality modules separately. In this paper, the cluster specific framework is defined to estimate the software quality.

Keywords: KMeans, Clustering, Software Modules, Software Fault, Fault Frequency, Software Quality

I. INTRODUCTION

Software quality is the primary aspect or the requirement for a software system. The quality of software system depends of multiple functional features, objectives and the aspects. These features can be terms of response time, processing time, failure probability, hardware support, platform support etc. To design a quality software system, the first requirement is to evaluate the quality requirement for software system and satisfy each of the dimensions. These dimensions are in terms of completeness, interpretability and accessibility vectors. The various aspects relative to software quality are shown in figure 1. These all aspects are associated to any individual module, process or the complete software project. The requirement specific, functional specific quality features are shown in the figure. To evaluate these features, the first requirement is to identify the type of module or the thing relative to which the quality estimation is performed. These aspects include the Compositive, Artifactitious, and Interactive forms. The quality phenomenon can be described by reusability, integrity, validity, reusability, testability, reliability, portability and maintainability. These aspects are low level aspects which can be classified to the efficiency, security, reliability and system support vectors. The evaluation of these measures can also done with specification of different integrated objects including the module estimation, component estimation, project estimation etc. Based on these aspects some measures can be applied

International Journal of Innovative Research in Science and Engineering
Vol. No.3, Issue 01, January 2017
www.ijirse.com
ISSN (O) 2454-9665
ISSN (P) 2454-0663

to evaluate the software quality. For real time applications and environment, more critical vectors represent the software system quality. These vectors are associated to software integrity and security. The software fault is one such vector can be quantified in different forms to represent the software quality.

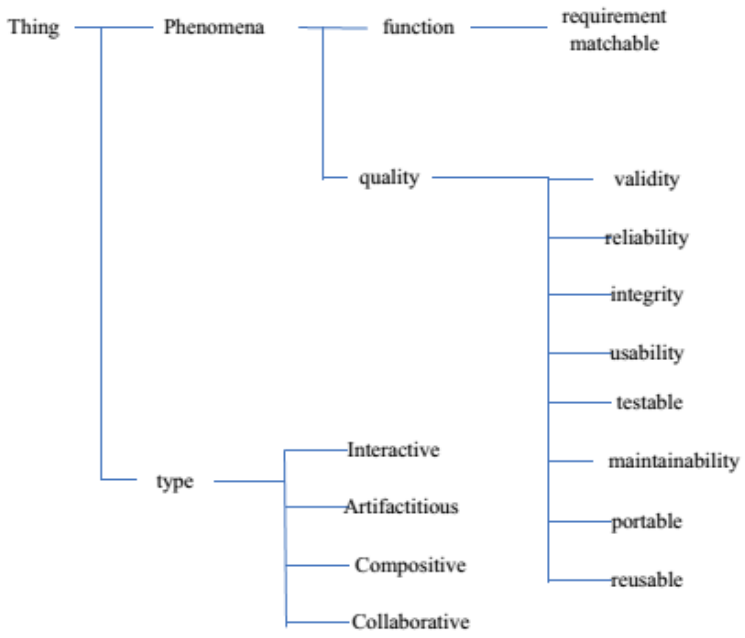

Figure 1 : Aspects of Software Quality

1.1 Software Fault

In a software system, fault can exist in different forms including the warning, errors and the failures. The severity of the software fault can be identified in terms of its impact on software system. Fault based software quality can be evaluated in terms of fault frequency, module criticality and fault criticality. A warning can be some error that will not match to the basic requirement or criteria but even then system will work fine. A fault can disrupt the system in terms of system performance, data loss or abnormal termination of any process. The failure can be a non fatal error that can fail the application or the system. A failure can gives heavy data loss or transactional termination. The module which relates the software fault and its importance also affects the software quality. The impact of that particular module on software system is also considered as the module-fault criticality. The number of faults occurs in a day or any relevant time period is defined as fault frequency. Higher the fault frequency, more critical the fault is considered. In this work, fault is considered as the main criteria based on which the software quality estimation is performed.

1.2 Clustering

Clustering is a data driven process that divides the available data in smaller segments called clusters. Some decision criteria are also defined to identify the cluster center and the cluster members. The distance specific analysis can be applied on data values to identify the clusters. The predictive decision can be taken based on this clustering approach. The clustering does not requirement any training set or some earlier class specification. It captures the current data features and processes them to generate the clusters. In this paper, the clustering method is used to identify the critical software modules based on different fault features. The fault weights of

ISSN (O) 2454-9665
ISSN (P) 2454-0663

these software modules are processed under clustering method to generate the clusters. The clustering view is shown in figure 2.

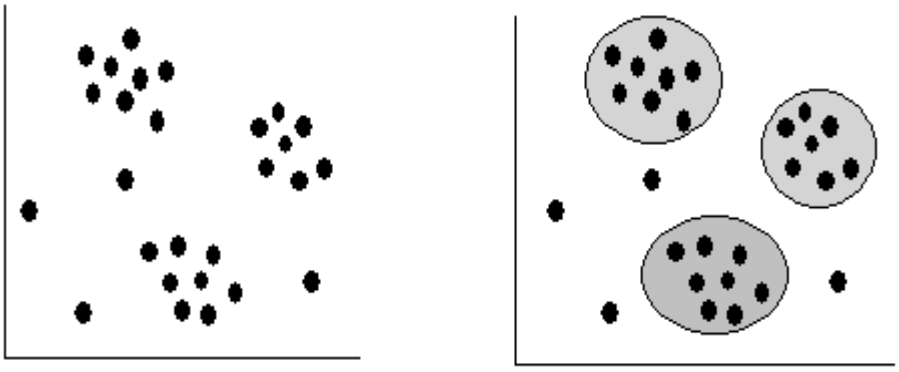

Figure 2: Clustering Behavior

Here figure 2 is showing the clustering behavior to divide the available data in smaller segments. The first figure is showing the randomly placed data values and second sub-figure is showing the formed clusters over the data. There are number of available clustering methods such as KMeans, CMeans or Hierarchical Clustering.

In this work, KMeans clustering is taken as main algorithmic approach to divide the software modules in clusters. Based on the cluster strength the overall quality of software system can be predicted. In this section, a wide description of various aspects of software quality is presented. The role of software fault to estimate the software quality is also described. The small description on clustering method is also provided in this section. In section II, the work provided by earlier researchers is discussed. In section III, the KMeans based parametric method is explored to analyze the software quality. In section IV, the conclusion of work is provided.

II.REVIEW OF LITERATURE

The primary requirement of software system is provide the quality solution for user requirement. There are number of vectors associated to system performance, risk and failures that affect the software system quality. In this section existing methods and features for evaluation of software quality are provided. Author[1] has provided a work on estimation of software cost under the quality aspect. Author used the fuzzy based measure for estimation of software quality. Author used the COCOMO1 and COCOMO II models for estimating the software quality. Eight different membership functions are defined by the author to estimate the quality. Error evaluation is also performed based on error rate observation. Another quality evaluation method based on the software quality and the featured aspect is provided by Author[2]. A decision support based module level evaluation is defined by the author. The multiple parameters are analyzed based on customer performance to estimate the software quality. The product level and process level quality measures are defined for the real time evaluation. Author[3] provided a work on performance evaluation as the major criteria for software effort evaluation. The cost and complexity specific analysis was provided to predict the software cost. The cost specific estimation is provided to identify the quality features. Once the features are identified, the cost evaluation is done. Based on this cost evaluation, the data software system categorization is done with estimation of quality aspects. Author[4] has estimated the software risk and measures it under fuzzy parameter.

ISSN (O) 2454-9665
ISSN (P) 2454-0663

The priority specific measures are applied by the author to analyze the system performance and integrity. The multistage model is defined to evaluate the software quality. Author[5] has defined static error analysis at code level to estimate the software quality. The cost and reliability based characterization is defined by the author for probabilistic evaluation. Author also identifies the mean failure rate so that the module and system level quality estimation can be done. The application specific measure can be applied to estimate the software quality.

Another complexity driven evaluation of software feature was provided by author[6] using fuzzy logic. The software system utilization and the mapping using binary decision tree is considered as the complexity criteria. The complexity series specification and its process in series and parallel are defined for quality estimation. A work on risk estimation and its mapping to the cost and quality vector is provided by author[7]. Author formulated the product level and process level estimation for development of effective software system. A fault tolerant software system was designed by the author with consideration of reliability aspect of software system. The environment specific software analysis was provided to estimate the software quality. Author[8] has used the fuzzy based method to estimate the software defect for software modules. A revised model was defined by the author to evaluate the defect rating for software system. The group maturity specification and software system organization is also provided by the author. A risk measurement based evaluation was provided by the author using Fuzzy logic. Author[9] measured the uncertainty aspect to measure the software quality and its evaluation using fuzzy logic. A reliability theory based evaluation was defined with probabilistic evaluation. A Markov model based featured estimation and transition was provided by the author to estimate the software quality. Author[10] has analyzed the software system under the statistical trends. The fuzzy specific estimation, statistical observation and cost evaluation was provided. The functional and interactive modular features are evaluated by the author. Author[11] has defined a decision making process for software system evaluation using fuzzy logic. A fuzzy preserved method was defined to generate the alternate solution. The fuzzy specific support to the reliability modeling was provided by the author. Decision feature estimation and software system evaluation was defined for effective system estimation. Author[12] has provided a featured evaluation on reliability aspect on feature estimation and evaluation. The fuzzy feature evaluation and mapping to the software quality was provided by author. The feature distribution and segment based estimation is defined to separate the error modules. Author[13] has defined a rule specific computational modeling to estimate the software risk fault and software quality. The multiple parameter based evaluation with comprehensive design was provided in this research.

III RESEARCH METHODOLOGY

The development of a quality product requires estimating various integrated vectors with each process stage of software system. The quality is itself a wider terms which can be defined in terms of software risk, error, performance and the dependency vectors. These vectors are already described in section I. In this work, fault feature evaluation based method is defined for software quality estimation. The integrated sub-features of software fault include fault type, fault frequency, fault criticality and the module criticality. Once the features are collected, these collected features are defined in the form of single quantitative weight. This individual weight value represents the features of each module. In the final stage, the KMeans clustering method is applied

ISSN (O) 2454-9665
ISSN (P) 2454-0663

on the featured weight value to divide the modules in different clusters. These clusters are representing the representing the set of modules based on the criticality observations. The flow of the work model is shown in figure 3

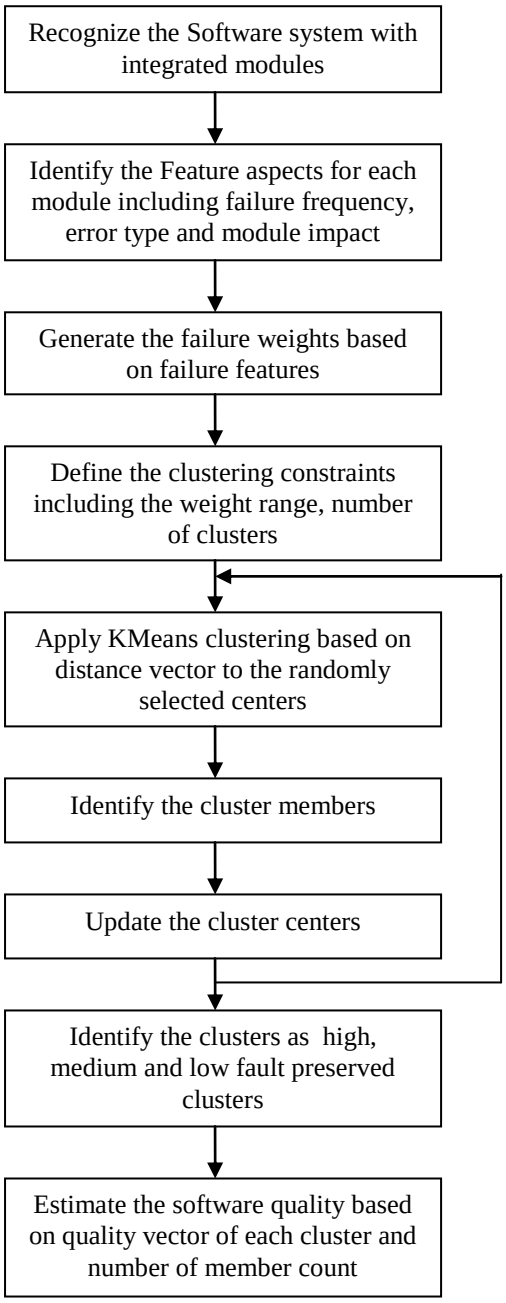

Figure 3 : KMeans based Software Quality Evaluation

Figure 3 is here showing the algorithmic model for dividing the software system in smaller clusters. The figure is showing the stage specific work. The input to the model is taken in terms of low level features of software modules. These features are related to the software system faults. These failure associated features includes the failure count, failure type, criticality and module criticality. Once the features are collected, these features are combined to generate the single weight value for each module. After setting up the dataset, the next work is to process the clustering algorithm. Clustering is a distance adaptive method to divide the modules based on weight

ISSN (O) 2454-9665
ISSN (P) 2454-0663

value. The process configuration requires the information about distance function and the number of clusters. The cluster center can be identified randomly. The distance based analysis is applied to identify the module members of all k clusters. Each time distance specific mapping of module weight to cluster center is done, the clustering results are obtained. After forming the clusters, the cluster centers are updated and the clusters are reform. This cluster updation process is repeated till the cluster members switch between the clusters. After obtaining the final clusters, the classification of these clusters based on the criticality vector can be done. Based on this cluster criticality and cluster member count, the quality of software system can be evaluated.

1.2 Clustering Algorithm

Clustering algorithm basically divides the available dataset in smaller segments or clusters. It is unsupervised learning method which does not require any class information. KMeans is one such known clustering method. In this method, the information about the number of clusters and the cluster center is required. After obtaining this information, the distance analysis is performed between the data element and each of centroids. The minimum distance cluster center will be considered as the as the cluster of that particular data element. With each of the successive iteration, the cluster center can be updated to generate more effective result. Kmeans clustering is one such efficient clustering method with low complexity. In this work, KMeans is defined as the decision algorithm to divide the modules in quality clusters. The algorithmic procedure for clustering is given hereunder

1. Define the number of clusters called K
2. Setup K random centroids over the data space
3. Perform the distance estimation of each of data members respective to the centroid. Define the distance estimation for each centroid
4. The centroid which is closed to the data member will be considered as parent to the data member.
5. Once the clustering is done, update the partitions based on nearest cluster based minimum distance estimation.
6. Repeat process, till the movement of data members between the clusters is performed.

IV. CONCLUSION

In this paper, the software quality estimation method is defined based on clustering method applied on fault features of software modules. The paper has defined the various aspects of software quality and software failure. The algorithmic method is also defined to use the clustering to identify the software system quality.

REFERENCES

[1] Ravishankar. S," Software Cost Estimation using Fuzzy Logic", International Conference on Recent Trends in Computational Methods, Communication and Controls (ICON3C 2012) Proceedings published in International Journal of Computer Applications

[2] B.K. Mohanty," Product classification in the Internet business—a fuzzy approach".

[3] S. Malathi," Performance Evaluation of Software Effort Estimation using Fuzzy Analogy based on Complexity", International Journal of Computer Applications (0975 – 8887)

ISSN (O) 2454-9665
ISSN (P) 2454-0663

[4] H. Iranmanesh," Risk Evaluation of Information Technology Projects Based on Fuzzy Analytic Hierarchal Process", World Academy of Science, Engineering and Technology

[5] Mikhail Glukhikh," Software Reliability Estimation Based on Static Error Detection".

[6] Harpreet Singh," Software Complexity for Computer Communication and Sensor Networks Using Binary Decision Diagrams".

[7] P. C. Jha," Fuzzy Approach for Selecting Optimal COTS Based Software Products Under Consensus Recovery Block Scheme", BVICAM's International Journal of Information Technology (BIJIT)

[8] A K Verma," Fuzzy Logic Based Revised Defect Rating for Software Lifecycle Performance Prediction Using GMR", BVICAM'S International Journal Of Information Technology

[9] F. JAVIER. CRESPO," On the Use of Fuzzy Regression in Parametric Software Estimation Models: Integrating Imprecision in COCOMO Cost Drivers".

[10] B. Praba," Fuzzy Reliability of Non Homogeneous Unified Fuzzy Possibilistic Markov Model", European Journal of Scientific Research ISSN 1450-216X

[11] S. Mahmoud Taheri," Trends in Fuzzy Statistics", AUSTRIAN JOURNAL OF STATISTICS

[12] Terry L. Hardy," MULTI-OBJECTIVE DECISION-MAKING UNDER UNCERTAINTY: FUZZY LOGIC METHODS".

[13] S. Sardar Donighi," A fuzzy reliability model for series-parallel systems", ISSN: 1735-5702