ESTIMATION OF SOFTWARE QUALITY PARAMETERS USING COMBINATION OF QUALITY FUNCTION DEPLOYMENT AND MESSY GENETIC ALGORITHM

HIMANSHU PANDEY
BBD University/Department of Computer Science, Lucknow,India
Email: hpandey010@gmail.com

DR. MANUJ DARBARI
BBD University/Department of Information Technology,Lucknow,India
Email: manuj_darbari@acm.org

BRIJESH VERMA
SVNIET/Department of Computer Science, Lucknow,India
Email: vermamtech05@gmail.com

GAURAV SRIVASTAVA
M.tech (Computer Science), BBD, Lucknow.
gaurav18hit@gmail.com

ABSTRACT: The paper explains the use of QGA (Quality Genetic Algorithm) functionality to be applied on the Usability and Complexity aspect of the software. It covers the independent runs of two teams in developing the website which is evaluated by separate group on the same guideline as QGA. The outcome of the two tests were analyzed and found to same.

KEYWORDS: Software Quality Parameters, QFD, Messy Genetic Algorithm.

INTRODUCTION

Software quality is gradually picking up now days. It is now days no more a luxury rather than an optional requirement. When a component, process or system fulfils the requirements, needs or expectations of customer then the degree is called SOFTWARE QUALITY. To increase the Quality and Reliability of the software it's need to be tested by various algorithms and methods. Achieving the quality standards of the software, we must have known the most incorporable factors and parameters of the software. These are the parameters that add up to the qualities of software are:-

Capability (functionality)

Functionality describes the amount of function which is contained by a product or software. The main purpose of developing software is to meet all the functional requirements. All the customer requirements and needs or business requirements is contained by functional requirements. Basically functional requirements define for what purpose software is being developed.

Usability

Software Usability is a combination of understandability, learn ability, operability and last the attractiveness of the product to the last user. Usability depends on these factors- comfort level, ease to use, simplicity etc.

Performance
By software performance we analyze the act of performing that software is work properly or not. These factors like communication failure, poor bandwidth, or components failure affected the performance of the software. The parameters for performance evaluation are- execution time, service unit reduction, idle time reduction etc.

Maintainability
Software Maintainability provides the capability to modify the software product. Modification of software contains corrections, improvements or adaptation to change in requirements and environments.

Durability
Software usability helps to amend the Software Durability. It has two parts- Data Durability and Session Durability. In order to increase the durability of the software we use technologies like Data Replication and Data Repair.

Serviceability
The ability to offer commit services by the software or application is called Serviceability . For the software it helps in terms of technical help, user manual and problem devolvement.

Availability
By availability we analyze of how probably the system is ready for use provides the repair or restart time into account.

Complexity
It has 2 types

a. *Apparent Complexity*- When a design or implementation is typical to comprehend and verify then that type of complexity is called Apparent Complexity.
b. *Inherent Complexity*- These are the factors like the number and intricacy of interfaces and number of conditional branches which are responsible for Inherent Complexity.

But one of the strongest criticisms of current customer is that they are unable to express their needs and expectations for the product or software. So it is very complicated to identify and list these most incorporable factors and parameters from various factors and parameters.

In Software Engineering, Quality Function Deployment is a method that capable to find those software parameters which plays major role to achieve the quality standard for software. By implementing the Quality function Deployment in various parameters and user requirements Software organizations become capable to identify the prioritized customer requirements and software parameters. But after applying QFD we are not able to identify all the software quality parameters and factors. In order to increase the percentage of identification of software quality parameters we used Messy Genetic Algorithm which is the variant of Genetic Algorithm.

In evolutionary algorithm Genetic Algorithm provides a multipurpose and most powerful optimization tool which starts from initial set of solution to derive new and possibly better solution.

Messy Genetic Algorithms (Goldberg et al. 1989) are well outfitted for software quality parameters selection problems. Messy Genetic Algorithms allow variable length strings that may be underspecified or over specified with respect to the problem being solved. A messy gene is a pair of two: Gene Number & Allele Value.

In order to mutate these parameters we use Messy Genetic Algorithm and by applying Quality Function Deployment we find out the actual gap between requirements of the customer and what can company provide.

The combination of Quality Function Deployment and Messy Genetic Algorithm help us in identify the appropriate software quality parameters to achieving the quality standard of the software and the combination of Quality Function Deployment would be a useful tool for managers of large software project.

RELATED WORK

The paper by Yonghua Zhou YuliuChen [6] on „QFD-based Decision-making Approach for Strategic BPR" derived house of quality to make the top-level decision-making strategy take both satisfying the requirement of external customer and settling the internal problems into account. We are inspired by the work of Hashem [1], M.M.A. which highlights the Global optimization through a new evolutionary algorithm. TIAN Na, CHE A-da [7] in his paper Goal Programming in Quality Function Deployment Using Genetic Algorithm clubbed the methods QFD and Genetic Algorithm. We have used this relation to identify the software quality parameters. K.Y. Chan1, T.S. Dillon1, C.K. Kwong2 and S.H. Ling [8] proposed that GP based method produce a more accurate and interpretable models than the other commonly used methods like QFD. They show this on his paper "Using Genetic Programming for Developing Relationship between Engineering Characteristics and Customer Requirements in New product". Norberto Eiji Nawa and Takeshi Furuhashi [2] used bacterial evolutionary algorithm to discover Fuzzy system parameters. The paper „Feature Selection and Clustering in Software Quality Prediction" by Qi Wang, Jie Zhu, Bo Yu [9] presents a new software quality prediction model based on genetic algorithm (GA) in which outlier detection and feature selection are executed simultaneously. Antonio Gonzalez and Francisco Hemera[3] worked on Iterative Rule Learning Approach in 1997. The concept of Fuzzy logic techniques are utilized in software reliability engineering. This concept is given by Xu, Z in 2001. Yi. Liu, Taghi M.Khoshgoftaar [12] in Genetic Programming Model for Software Quality Classification compared the two methodologies LRM and GP and shows that GP model is much better than LRM. . S. Keshavarz and Reza Javidan [13] deals with Software Quality Control with criteria of covering application and proposed a new method based on genetic algorithm for generating optimal test data. In this paper „Comparison of Software Quality Models: An Analytical Approach" Sanjay

Kumar Dubey, Soumi Ghosh, Prof. (Dr.) Ajay Rana [11] analyzes the qualitative characteristics and side-by-side determines the software quality. Paper by Salah Bouktif, Bal´azsK´egl, HouariSahraouiIn [14] combines Software Quality Predictive Models. We are inspired by the work on Combining Models because combined models works well on the particular system or in the particular type of organization. David A and Gary B. Lamont [5] focus on Messy Genetic

Algorithms for Multi Objective Optimization. We are inspired by the work of Linda Murphy, Hoda S. Abdel-Aty-Zohd, M.Hashem-Sherif [10] they tracked the Genetic Algorithm Model for Product Deployment in Telecom Services. They have used the genetic algorithm to update the parameters applied to the input measurements to find the optimum solution for the defect tracking model system. Hillol Kargupta [15] introduces the gene expression messy genetic algorithm (GEMGA). It directly searches for relations among the member of the search space. We have used the Messy Genetic Algorithm to find the appropriate customer requirements and engineering characteristic in our paper. Lastly the paper by Kohei Arai [16] compares the conventional simple Genetic Algorithm to Messy Genetic Algorithm. This highlights the importance of Messy Genetic Algorithm for clustering. We have extended the concept of indexing from messy genetic algorithm to identify the highly prioritized customer requirements and engineering characters.

OUR FRAMEWORK

To highlight the Quality Parameters we have formed two student's team developing Module-1 and Module-2. These teams are assigned a task of developing a website of educational institute. They are given guidelines about the basic functionality which is to be embedded in the website. These are basically selected to navigational guidance like providing the basic user interface and task selected documentation. The basic categorization of the parameters is: (adapted from Rick Sobiesiak and Tim O keefe) [17].

Context Shift
It refers to paradigm shift when a user changes from standalone systems to Web based system. (i.e. how easily it can migrate from one system to another.)

Navigational Guidance
It refers to the support provided to user on step-by-step basis. Like Intellisense in Microsoft.

System Feedback
It is an instant feedback which the user service while performing any action.

Error Feedback
It is the system's response when user encounters an error. It provides help information and troubleshooting guidance.

These four categories of Voice of Customers are linked with Usability and Reliability features which the software companies can offer. In order to link between the voice of customer (Customer"s Requirement) and what the company can provide.

We have used Quality Function Deployment providing a relationship between "WHAT" the customer wants and "HOW" the software industry will achieve. We start with finding out the explicit and implicit customer requirements and write it on the Horizontal Partitions of QFD. On the contrary the "Technical Factor's" which the company can offer are given by Vertical Partitioning of features like "Usability" and "Reliability" etc.

The relationship can be given by developing the "Competitive Assessments" which includes customer competitive assessment and Technical Competitive assessment which the company can give in their product or do necessary amendments.

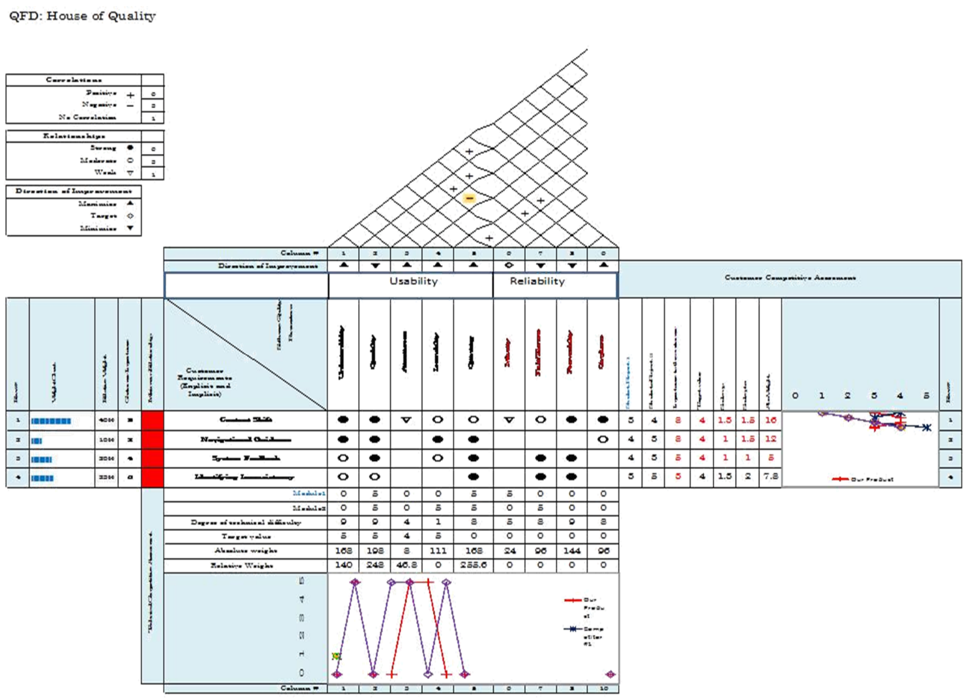

Figure1. QFD

From Figure.1 we are unable to sort out the Voice of Customer and Prioritized Technical Descriptor, under voice of customers. “Context Shift” and “Navigational Guidance” is having the highest score so we select these two as the highest priority factors. Similarly “Understandability”, “Operability” and “Optimizing” features are taken as high priority delivery. We can write them as two sets:

Understandability	Operability	Optimizing

Prioritized Technical Descriptor

Context Shift	Navigational Guidance

Prioritized Voice of Customer

Figure 2 Two sets of Descriptors

Figure 2 represents the two set of descriptors: Prioritized Voice of Customer and Prioritized Technical Descriptor. Context shift and Navigational Guidance having highest score in the QFD so take it is as a set of Prioritized Voice of Customer. Understandability, Operability and Optimizing having highest score in the QFD so take it is as a set of Prioritized Technical Descriptor. To find out the best combination of voice of customer and prioritized technical descriptor we use Messy Genetic Algorithm feature. The reason for using Messy Genetic Algorithm is it uses variable length and position independent coding. Even if we interchange the sequence of the two parameters the output remains unaffected.

The two Teams 1 and 2 made the website of a university based on the "Information Flow" and "Usability". It was given to an independent student's team to rate it according to various parameters given in the Questionnaire.

Team 1: Website having Excellent Navigational Capability

Team 2: Website with Poor Navigational Capability

Figure3. Comparison of website pages makes by Teams 1 & 2

Figure 3 shows a comparison of website pages makes by student Team 1 & 2. Team 1 has taken the result of QFD and applied this result into the website thus the Website made having Excellent Navigational Capability and it's satisfying the 3 click rule. While Team 2 has not applied the result of QFD into the website thus the Website has a Poor Navigational Capability and it's not satisfying the 3 click rule.

ANALYSIS

Starting with the Genetic QFD phase we applied the relationship given between "Voice of Customer" and "Prioritized Technical Descriptors" into set of relational rules. For e.g.: "Understandability" and its relationship with "Context Shift" and "Navigational Guidance" under one set of Rule Chromosome.

Understandability	UC	UN	UO	UO
Operability	ON		OC	
Optimizing	OC		ON	
Context Shift	………………			
Navigational Guidance				

Rule 1	Rule 2	…	Rule N

Figure 4**.** Rule defining the ratio of Relationship in Fuzzy Linguistic terms

Figure 4 represents the rule defining the ratio of Relationship in Fuzzy Linguistic terms. Figure shows the relationship between Voice of Customer and Prioritized Technical Descriptors into set of relational rules.

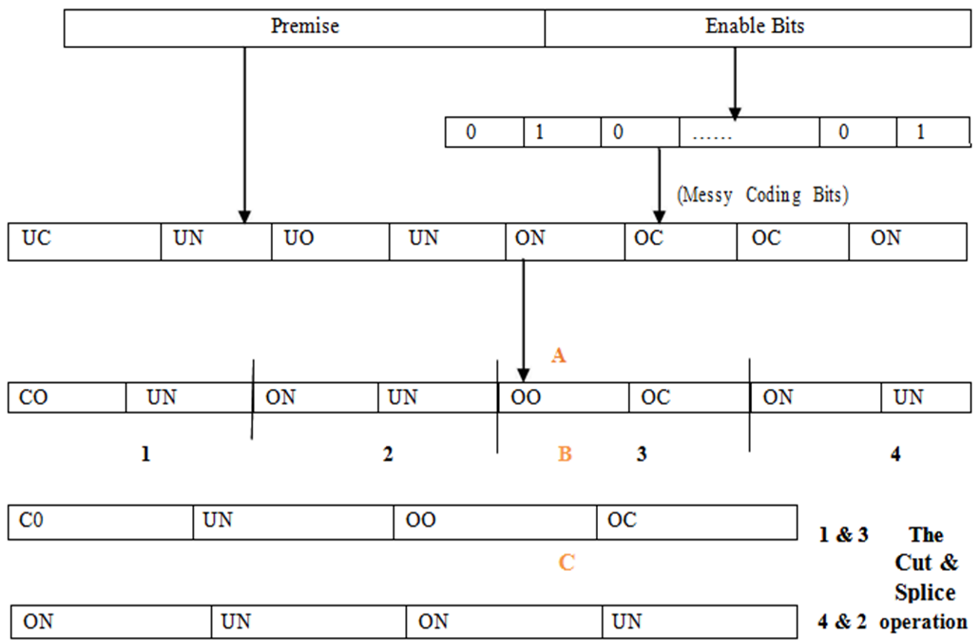

Figure5. Messy Coding and positional preferences using Cut and Splice operation

Figure 5 defines the Messy Coding and positional preferences. Messy coding used a Cut and Splice operation for successive mutation.

The key idea is to generate an index to the gene allowing the identity of its position. The crossover operator is replaced by more general cut and splice operator allowing the parents to make with different lengths. The idea is to cut and splice for both parents independently and splice the four fragments. After successive mutations it was found that "Understandability" and "Navigational Guidance" emerged as the most perfect outcome of Messy Algorithm.

From the Questionnaire and Stream listing of the websites, we found that Team 1 was rated higher than Team 2. The Parameters on which the teams give due weightage were "Understandability" and "Navigational Guidance" in term of Ease of Use.

Thus we can say that the results generated using QFD and Messy Algorithm exactly matched-up with the independent test conducted on student's team.

CONCLUSION

The paper highlights the issue of that determines the Gap Analysis between "WHAT" the customer wants and "HOW" the software development company can achieve it. During the first stage a framework is developed using QFD and Messy Genetic Algorithm which selects the best possible combination of what the customer wants and how the company can achieve it. Secondly in the second stage it selects the independent team to analyze the Model Website developed by two teams. The outcome of both the tests helped in determining the highest priority parameters of both the sets i.e. "WHAT" and "HOW". In future we will be developing a framework which will use multi objective Messy Systems evolving to a more accurate result.

REFERENCE

Hashem, M.M.A: "*Global Optimization Through a new class of Evolutionary Algorithms*", PhD Dissertation, Saga University, Japan, 1999.

Norberto Eiji Nawa and Takeshi Furuhashi, "*Fuzzy system Parameters. Discovery by bacterial Evolutionary Algorithm*", IEEE Transaction on Fuzzy system. Volume 7, No5, 1991.

Antonio Gonzalez and Francisco Hemera , "*Multi- stage Genetic fuzzy systems based on the Iterative Rule Learning Approach"* , Math ware & Soft Computing 4(1997).

Microsoft Corporation, "*Usability in Software Design"*. A while paper by Microsoft, October 2000.

David A and Gary B. Lamont, "*Multi Objective Optimization with Messy Genetic Algorithm*", S.Ac, 2000 Italy.

Yonghua Zhou Yuliu Chen, "*The QFD-based Decision-making Approach for Strategic BPR*" Beijing 100084, P. R. China.

TIAN Na, CHE A-da, "*Goal Programming in Quality Function Deployment Using Genetic Algorithm*", International Conference on Management Science & Engineering August 2007.

K.Y. Chan1, T.S. Dillon1, C.K. Kwong2 and S.H. Ling, "*Using Genetic Programming for Developing Relationship between Engineering Characteristics and Customer Requirements in New Products*" , Digital Ecosystems and Business Intelligence Institute, Curtin University of Technology, Perth, Australia.

Qi Wang, Jie Zhu, Bo Yu, "*Feature Selection and Clustering in Software Quality Prediction*", Evaluation and Assessment in Software Engineering,2007.

Linda Murphy, Hoda S. Abdel-Aty-Zohd , M.Hashem-Sherif, "*A Genetic Algorithm Tracking Model For Product Deployment in Telecom Services"*, 0-7803-9197-7/05/© 2005, IEEE.

Sanjay Kumar Dubey, Soumi Ghosh, Prof. (Dr.) Ajay Rana, "*Comparison of Software Quality Models: An Analytical Approach*", ISSN 2250-2459, Volume 2, Issue 2, February 2012.

Yi. Liu, Taghi M.Khoshgoftaar , "Genetic *Programming Model for Software Quality Classification*", 6th IEEE International Symposium on High Assurance Systems Engineering (HASE"01) 2001.

S. Keshavarz and Reza Javidan, "*Software Quality Control Based on Genetic Algorithm*", Vol. 3, No. 4, August 2011.

Salah Bouktif, Bal´azs K´egl, Houari Sahraoui, "*Combining Software Quality Predictive Models: An Evolutionary Approach*"**,** Dept. of Computer Science and Op. Res., University of Montreal C.P. 6128 Succ. Centre-Ville, Canada.

Hillol Kargupta, "*The Gene Expression Messy Genetic Algorithm*", published in IEEE conference on Evolutionary Computation, Nagoya, Japan, 1996.

Kohei Arai, "*Clustering Method Based on Messy Genetic Algorithm: GA for Remote Sensing Satellite Image Classifications*", International Journal of Advanced Research in Artificial Intelligence, Vol. 1, No. 8, 2012.

Rick Sobiesiak and Tim, "*Complexity analysis: a quantitative approach to usability engineering*", IBM Rochester Laboratory.

M Darbari, N Dhanda "*Applying Constraints in Model Driven Knowledge Representation Framework", International Journal of Hybrid Information Technology,2010.*

Development of effective Urban Road Traffic Management using workflow wechniques for upcoming metro cities like Lucknow (India), M Darbari, S Medhavi, AK Srivastava , Development 2 (2)4,2008.

Application of Use Case for Identification of Root Cause of the Dependencies and Mutual Understanding and Cooperation Difficulties in Software Systems, IA Siddiqui, M Darbari, International Journal of Applied Software Engineering, 4, 10-20.

IA Siddiqui, M Darbari, S Bansal , "*Application of Activity Theory and Particle Swarm Optimization Technique in Cooperative Software Development*" ,International Review on Computers & Software 7 (5)

IA Siddiqui, M Darbari, "*A Group Awareness and collaboration in Distributed Software Development*", International Journal of Scientific & Engineering Research Volume 3, Issue 3, March 2012.

M. Darbari, H. Pandey, V. K Singh, "*Coalescence of Evolutionary Multi-Objective Decision making approach and Genetic Programming for Selection of Software Quality Parameter"*, International Journal of Applied Information System 7(11):18-22, November 2014.

J. Verma, S. Bansal, H. Pandey, *"Develop Framework for selecting best Software Development Methodology"* ,International Journal of Scientific and Engineering Research. Volume 5, Issue 4, April 2014.

M. Srivastava, H. Pandey, and B. Thakur, "*A Literature Review of E- Learning Model Based on Semantic Web Technology"***,** International Journal of Scientific and Engineering Research" Volume 5, Issue 10, October 2014.