INTELLIGENT SOFTWARE QUALITY MODEL: THE THEORETICAL FRAMEWORK

Jamaiah Yahaya[1], Aziz Deraman[2], Siti Sakira Kamaruddin[3], Ruzita Ahmad[4]

[1]*Universiti Utara Malaysia, Malaysia, jamaiah@uum.edu.my*
[2]*Universiti Malaysia Terengganu, Malaysia, a.d@umt.edu.my*
[3]*Universiti Utara Malaysia, Malaysia, sakira@uum.edu.my*
[4]*Universiti Utara Malaysia, Malaysia, rita_azura@yahoo.com*

ABSTRACT. Globally, software quality issues has increasingly been seen as a common strategic response for achieving competitiveness in business. It has been seen very important as the usage of software become very demanding. Software quality includes quality control tests, quality assurance and quality management. Currently, software quality models available were built based on static measurements of attributes and measures. Previous study has indicated that to ensure the quality of software meets the future requirements and needs, the new dynamic and intelligent software quality model has to be developed. This paper discusses the development of intelligent software quality model based on behavioral and human perspectives approach which enhances from Pragmatic Quality Factor (PQF) model as a benchmark of the quality assessment.

Keywords: software quality model, intelligent quality model, software assessment, theoretical framework

INTRODUCTION

Nowadays, rapid development and diffusion of software quality and ICT related technologies in several industries worldwide, shows that the statistics for insufficiently understood requirements account pointed to 50% of all errors and followed by design incorrectly understood or incorrectly translated from requirements account for 30% of all errors. Hence, the programming error or misunderstood of system design contributes to 20% of all errors. Thus, Software Quality Assurance (SQA) is very important issue in software development for reducing the rate and associated cost of failure through poor product and services (Bevan, 1999).

In many organizations, software is considered as one of the main assets with which the organization can enhance its competitive global positioning. At the same time, software house companies are competing to produce software which are claimed to be good and fulfill user's expectation and requirements. The companies unable to provide any justification on the quality of their products to the users and users are left with uncertainties on the standard and quality of the software. Even though many quality models have been developed by several researchers, this issue is still in debate and the quality of software products are still questioned. It is because quality is believed as the eyes of the beholder. This paper presents the new software quality model with an intelligent aspect integrated and relevant in current and future requirements. It presents the problem statement, research objective, literature review and methodology. The last section concludes the paper.

PROBLEM STATEMENT

Currently, the existing quality models have consistently highlighted as static models which have been develop to measure software product in term of time, budget and also correctness with efficiency to perform all the specified functions of requirements (Denning, 1992). However, these models have been developed to evaluate and measure software product based on technical aspects and theories but did not focused on human aspect (Yahaya, Deraman & Hamdan, 2008).

Software quality environment today is seems as fast changing in term of users requirements and needs to fulfill for current and future requirement in assessment. Previous study indicated that new attributes might be needed to be included in the assessment as well as in the model (Yahaya, Deraman & Hamdan, 2006). The new attributes could not be added in the current static quality models. The two problems mentioned motivates us to develop an intelligent software quality model with the new dynamic tool and technique by highlighting to the human aspects. As to control and maintain the quality of software product, this proposed dynamic model should capable to identify and recommend to the environment the new behavioral attributes which supposed to be maintained or included in the model. The proposed intelligent approach is based on Pragmatic Quality Factor (PQF) model which was developed by Yahaya, Deraman and Hamdan (2008). It will be known as a new intelligent software product quality model which includes the multi intelligent aspect in this model. The original version of PQF model does not include intelligent technique and based on static method.

RESEARCH OBJECTIVE

The main purpose of this research is to construct an intelligent Pragmatic Quality Factor (i-PQF) model with the proposed AI technique. This will be supported by the following objectives which are to investigate the quality and assessment problem in software product, to develop a theoretical framework for intelligent software quality model, to proposed Artificial Intelligence technique for software quality model based on PQF model and to test the model by prototype.

The scope of the study is to develop the intelligent software quality model based on the software behavioral and human factor by using Artificial Intelligence technique. Thus, this research is developing the quality model in multi intelligent aspect and to propose some enhancements in the original PQF model.

LITERATURE REVIEW

Software quality is defined as a set of quality attributes with different interest group express their needs of software. International Organization for Standardization (ISO) defined quality as the totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs (Yamada, 1996). ISO/IEC9126 (1996) applies software quality as to extend the user needs are well defined. Hence, IEEE (1993) defines software quality as a software features or characteristics to assess the quality of a system or component. Software quality is also defined as the fitness for use of the software product and conformance to software requirements and to provide useful services (Tervonen, 1996). Thus, software quality also is defined as conformance to explicitly stated functional and performance requirements, explicitly documented development standards, and implicit characteristics that are expected for all professionally developed software (Tervonen, 1996). In many organizations, software is considered as one of the main assets with which the organization can enhance its competitive global positioning in current economic era. The different perspectives of quality were referred to the transcendental view, user view, manufacturing view and value-based view. These perspectives influence the software product quality development in term of their final interaction with the final product.

The relationship of each quality attributes in software product provides an abstract or high level specification for the software product and the capability to fulfill the changes in the environment is important to be highlighted. Several software product quality models are available from literature and the most well known models are McCall, Boehm, FURPS, ISO9126, SQuaRE, Dromey, Systemic and Pragmatic Quality Factor (PQF) model. Each of these models measure and evaluate the quality attributes of software products from various dimension with different characteristics. It shows that the main quality characteristics found in majority of these models are: efficiency, reliability, maintainability, portability, usability and functionality, which are presented in more recent models. Most of these models are only focused on developer view rather than user's view in measuring and evaluating the quality attributes in software product. The existing model has been constructed with conformance to requirements and does not to focus on user expectations and satisfactions with including the theoretical and technical perspectives. Thus, these static models are lack of capability to fulfill the future requirements in this volatile environment (Yahaya & Deraman, 2010).

The development of PQF was intended to evaluate and measure software quality attributes by fully highlighted on software behavioral and human aspects. This new static model has showed and clearly defined the way to evaluate and measure quality attributes as well and improved the lack of mechanism and techniques of software assessment in existing models. It covered the behavioral aspects of the software and the impact attributes. This model supports a better software quality standard and procedures to fulfill user requirement and user needs. PQF has been applied in real industrial practices with involve several large organizations in Malaysia (Yahaya, Deraman & Hamdan, 2007; 2010).

From literature, several studies were implemented in software quality projects using Artificial Intelligence techniques for several purposes. Some of the studies were by Briand et al. (2000), Burgess et al (2000), Goa, Khoshgoftaar & Wang (2009) and Aguero et al. (2010). Several methods and techniques in AI approaches such as Feature Selection (FS), Artificial Neural Network (ANN) and Cased-Based Reasoning (CBR) has been investigated and reviewed. Feature Selection (FS) is a process of selecting a subset of relevant features for building learning models and it used to remove less important features from the training data set (Gao, Khoshgoftaar & Hulse, 2010). Feature Selection Wrapper-Based Feature Ranking technique which is a part of Feature Selection includes wrapper approach to assess the attributes individually and ranks the attributes according to their individual predictive power. Whilst, Artificial Neural Network (ANN) technique is created by presenting the network with inputs from many records and the function in this technique is not able to be used in calculating the weight of each attributes. The same reason also goes to another technique which is Case-Based Reasoning (CBR). CBR only adapts previous stored solutions that can been found to be effective in the solution of earlier problems (Khoshgoftaar et al., 2003). Therefore, CBR is not suitable to be used in construction the algorithm in our proposed intelligent quality model. In addition, both techniques are lack of capability to fulfill the current needs and the future requirement in development of the proposed quality model.

From the literature study, the Automatic Hybrid Search (AHS) and Hybrid Feature Selection (HFS) are the methods which using the wrapper approach in their processing function and both methods have the capability in assessing and ranking each of the characteristics by their predictive power (Gao, Khoshgoftaar & Hulse, 2010). The wrapper approach is using the same learning algorithm that will be used for learning or classifying the domain represented with the selected feature subset for example Naïve Bayes, K-Nearest Neighbour (KNN), Support Vector Machine (SVM) or Logistic Regression (LR) (Khoshgoftaar et al. 2003). Both methods are suitable to be used compared to another proposed methods such as Kolmogorov-Smirnov (KS), Probabilistic Search (PS) and Rough Set (RS) due to it relevants to create the new algorithm as needed and achieved the objectives in this study.

METHODOLOGY

The research will be carried out in five main phases with the aim is to develop a new intelligent software quality model. The phases are:-

Theoretical Study

The purpose of this phase is to investigate the quality and assessment problems in the existing software product. The inputs from this phase will be books, journals, proceedings and any other documentations related to software quality. The deliverable from the study can support the researcher to generate the new ideas on identifying software quality attributes and models as well as their strengths and weaknesses in measuring the attributes.

Design of formal framework on intelligent software quality

This phase involves identifying the specific feature of software quality to be represented using AI approach. The formal framework can help the researcher to determine the study areas and also content considerations, research questions that need to be addressed and the methodology. Figure 1 illustrates the theoretical framework on intelligent software quality model.

The development of i-PQF will involve several components of static models by focusing the additional human aspects and multiple intelligent aspects. Two main issues will be studied which by means of model and component. Under these two issues, the details of available and relevant software quality models and associated components will be discovered. The third issues will be the dynamic aspect of software quality. This will cover the intelligence techniques and methods in developing the intelligent software quality model. Thus, the construction of the new dynamic and intelligent model is based on the PQF model as the based line in attributes assessment. The development of prototype will be used as a medium for running the proposed algorithm.

Identify and propose the AI technique for intelligent software quality model

Several methods and techniques will be studied in order to enhance the software quality model by using AI approach. The methods and techniques are described in previous section. In order to create the new algorithm with the intelligent function to measure the attributes in software quality, the Feature Selection Wrapper-Based Feature Ranking technique has been chosen to be applied in this model. Furthermore, this chosen technique also contained several methods to be integrated in the new algorithms which are Automatic Hybrid Search (AHS) and Hybrid Feature Selection (HFS).

Construction of intelligent software quality (i-PQF) and prototype

In this new intelligent model, the collection of the data will be inherited from the PQF model which is based on Likert scale of values 1 to 5. In addition to the collected data, PQF consists of ranking values of weighted scale that being assigned by the users to evaluate software product. Consequently, the data will be compiled by using the classifier such as C4.5, MultiLayer Perceptron (MLP), Naïve Bayes (NB), or Logistic Regression (LR) which are already included in WEKA tool. Thus, the concept, definition and the characteristics of the attributes still be inherited and the additional concept of intelligent system by using AI approach will be added. The algorithm will be integrated and embedded in a prototype that will be developed in this phase. During the experimental design on the algorithm, three main steps will be implemented: input, data processing and testing.

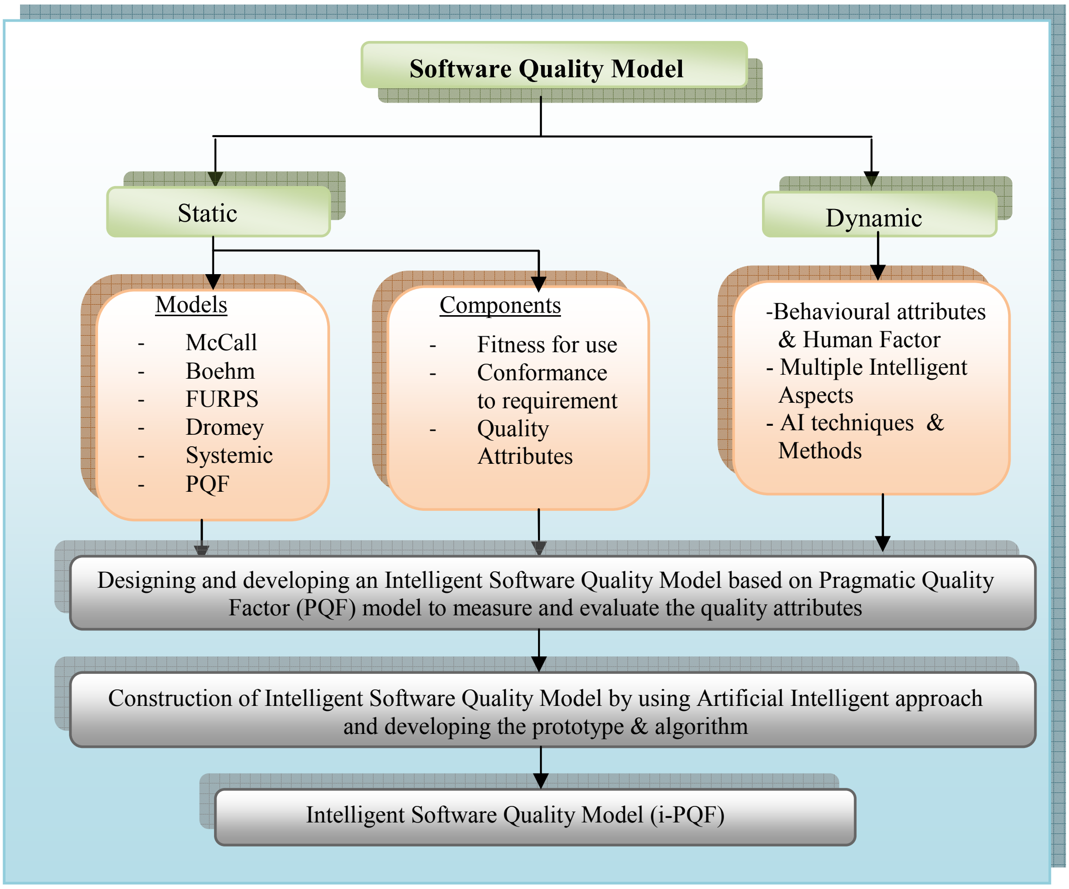
Software Quality Model
Static
Dynamic
Models
- McCall
- Boehm
- FURPS
- Dromey
- Systemic
- PQF
Components
- Fitness for use
- Conformance to requirement
- Quality Attributes
-Behavioural attributes & Human Factor
- Multiple Intelligent Aspects
- AI techniques & Methods
Designing and developing an Intelligent Software Quality Model based on Pragmatic Quality Factor (PQF) model to measure and evaluate the quality attributes
Construction of Intelligent Software Quality Model by using Artificial Intelligent approach and developing the prototype & algorithm
Intelligent Software Quality Model (i-PQF)

Figure 1. The Theoretical Framework on Intelligent Software Quality

Confirmation Study

In this phase the proposed model and the prototype will be tested and validated. In the testing process, two types of testing will be carried out which are alpha testing and beta testing. The testing will be done on the prototype and the new proposed algorithm will also be tested and validated to confirm the functionality of the new intelligent model.

CONCLUSION

The criteria of software assessment in software quality might change and always require additional new criteria to be included in future. The idea to enhance the static model to a dynamic model in software quality by using PQF model as a standard model become the main focus to be highlighted in this study. Thus, the intelligent toolset should be capable of adapting and noticing any changes in the environment. This can be done by integrating Artificial Intelligence techniques and methods. Furthermore, by attaching the intelligent system criteria in this new dynamic quality model can support the certification environment with a self-learning capability with capturing knowledge from previous certification processes and experiences. Thus, the new model can be used as a benchmark for another researches that relate to software quality and certification.

ACKNOWLEDGMENTS

The research is funded by the Fundamental Research Grant Scheme, Ministry of Higher Education, Malaysia.

REFERENCES

Aguero, M., Madou, F., Esperon, G. & Lopez, D. L. (2010). Artificial Intelligence for Quality Improvement. *World Academy of Science and Technology,* 63.

Bevan, N. (1999). Quality in use: Meeting user needs for quality. *The Journal of Systems and Software,* 49, March 19, 89-96.

Briand, L. et al., (2000). Exploring the relationships between design measures and software quality in object-oriented systems. *Journal of Systems and Software,* 51, 245–273.

Burgess, C.J., (2000). Using Artificial Intelligence to Solve Problems in Software Quality Management. *The 8th International Conference on Software Quality Management(SQM2000), Software Quality Management VIII.* ISBN 1-902505-25-5, 77–89.

Denning, P.J. (1992). What is software quality? *A Commentary from Communications of ACM (January).*

Gao, K., Khoshgoftaar, T. & Hulse, J.V. (2010). An evaluation of sampling on filter-based feature selection methods. *Proceedings of the Twenty-Third International Florida Artificial Intelligence Research Society Conference (FLAIRS)*, 416-421.

Gao, K., Khoshgoftaar, T.M. & Wang, H. (2009). An Empirical Investigation of Filter Attribute Selection Technique for Software Quality Classification. *Proceedings of The 2009 IEEE International Conference in Information Reuse and Integration, Las Vegas, Nevada, USA.*

IEEE. (1993). IEEE standard for a software quality metrics methodology. Retrieved August 20, 2010 from http://ieeexplore.ieee.org/xpl/standards.jsp.

ISO/IEC 9126. (1996). Software quality characteristics and metrics-Part2: External metrics. *Technical Report,* ISO/IEC JTC1/SC7/WG6.

Khoshgoftaar, T.M., Nguyen, L., Gao, K. & Rajeevalochanam, J. (2003). Application of an attribute selection method to CBR-based software quality classification. *IEEE Software,* 47-52.

Tervonen, I. (1996). Support for quality-based design and inspection. *IEEE Software (January),* 44-54.

Yahaya, J.H., Deraman, A. & Hamdan, A.R. (2006). Software Quality and Certification: Perception and practices in Malaysia. *Journal of ICT (JICT),* **5**(Dec), 63-82.

Yahaya, J.H., Deraman, A. & Hamdan, A.R. (2007). A case study in applying software certification model by product quality approach. *Proceedings of The International Conference on electrical Engineering and Informatics, June 17-19, Bandung, Indonesia,* 706-709.

Yahaya, J.H, Deraman, A. & Hamdan, A.R. (2008). Software quality from behavioural and human perspectives. *IJCSNS International Journal of Computer Science and Network Security,* 8(8), August 30, 53-63.

Yahaya, J.H, Deraman, A. & Hamdan, A.R. (2010). Continuously ensuring quality through software certification: A case study. *Proceedings of The International Conference on Information Society (i-Society 2010), June 28-30, London, UK.*

Yahaya, J.H. & Deraman, A. (2010). Measuring the Unmeasurable Characteristics of Software Product Quality. *International Journal of Advancements in Computing Technology (IJACT),* 2(4),95-106.

Yamada, A. (1996). Information Technology – Software quality characteristics and metrics Part 2 – external Metrics”, *Draft Technical Report,* ISO/IEC JTC1/SC7.