

# MODELING SOFTWARE QUALITY EVALUATION USING THE ANALYTIC HIERARCHY PROCESS

Min-Suk YOON\*

Department of Electronic Commerce, Chonnam National University  
Yeosu, Chonnam, S. Korea  
E-mail: [msyoon@chonnam.ac.kr](mailto:msyoon@chonnam.ac.kr)

Buhm Lee

Department of Electrical and Semiconductor Engineering, Chonnam National University  
Yeosu, S. Korea  
E-mail: [buhmlee@chonnam.ac.kr](mailto:buhmlee@chonnam.ac.kr)

Kyu-Seung WHANG

Business Administration, Korea University  
Seoul, S. Korea  
E-mail: [kswhang@korea.ac.kr](mailto:kswhang@korea.ac.kr)

## ABSTRACT

This paper proposes a modified software quality evaluation method based on products comparison, judging by user perceived satisfaction in terms of quality. The focused evaluation model is formulated on the bases of the ISO 9126 quality model as a generally accepted standard and ISO 12119 for software package, and of the Analytic Hierarchy Process (AHP) which is appropriate for comparing paired objects with subjective judgment. Getting priorities or weights vector, the AHP/ANP (Analytic Network Process) is usefully accepted when the evaluation depends on subjective judgment and/or expert knowledge. An outstanding advantage of the AHP/ANP is pairwise comparisons, which can covers subjective and fuzzy evaluation. However, it is not uncommon that an evaluator does not empirically experience all of multiple software products in a same category and that paired comparison is known to be more valid than independently single alternative evaluation. This leads to the limited pairwise comparisons among alternatives from each evaluator's experience and judgment even though the method is the way to raise the comparison validity. Then, the required pairwise comparison matrix is proposed by gathering such limited comparisons, called partial pairwise comparison matrix, from all evaluators who has different products experience or knowledge. The proposed model is empirically exemplified with popular software products. Only reliable data, identified by screening with the consistency ratio of the AHP, are analyzed. This paper discusses to estimate missing value and rank order with the result using the compatibility index in the AHP.

**Keywords:** Software Evaluation Model/Method, Analytical Hierarchy Process, Pooling Partial Pairwise Comparisons; Missing Value Estimation, Compatibility Index

## 1. Introduction

This study proposes appropriate way to evaluate software products using comparisons, especially in terms of user perceived quality. For the past decades, since software technology has been developed very much, more and more software has been produced and distributed as package and software competition has

---

\* Corresponding author

intensified. Software is defined as programs, procedures, rules and any associated documentation pertaining to the operation of a data processing system [ISO 8402], and software product is a software entity designated for delivery to a user. A software package is a complete and documented set of programs supplied to end users for a generic application or function [ISO 12119]. Application packages are sold through off-line or on-line channels. This has created a difficult problem of software evaluation and choice caused by the multiplicity of competing products-

Quality is defined as the totality of characteristics of an entity that bears on its ability to satisfy stated and implied needs [ISO 8402] and it is a driver for user satisfaction. As a result, the definition of quality to satisfy user needs is transferred to software product consistently [ISO 9126]<sup>1</sup>. Software-metrics advocates assume that measuring and controlling internal product properties (internal quality) will result in improved external product and quality in use.

We adopt software quality as comprehensive concept to end-users. In evaluating software from user view, there are two issues to point out. First, it is not usual that the user has experienced over all products that are focused. Second, when individual product evaluation is suggested, it usually requires the validity or justification of the result, which is usually concerned with the number of evaluators.

The objective of this study is to propose a method that evaluates user-perceived quality of software, representing usage characteristics. Additionally, this model shall not only identify the dominant criteria as key drivers of user-perceived software quality but also extract weak and strong points contributing to quality preference in the competitive market. The second objective of this study lies in the implementation of the model.

## **2. Quality Criteria and Model**

### **2.1. Software Quality Criteria**

The judgment through end users' perception replaces most objective metrics that will be transformed into the description in user language. Quality in information systems is a multi-dimensional concept [Curtis, 1980; Cusumano and Kermerer, 1990]. Likewise is quality in software products. Understanding or measuring software quality has been discussed through the hierarchical quality model. A quality model is defined as the set of characteristics and the relations between them which provide the basis for specifying quality requirements and evaluating quality [ISO 9126]. More precisely, quality evaluation is the systematic examination of the extent to which an entity is capable of fulfilling specified requirements. A quality characteristic is a set of attributes of a product by which its quality is described and evaluated. An attribute is a measurable physical or abstract property of an entity [Fenton, 1994]. The quality attributes may be conflicting or cooperative among themselves [ISO 14598-1].

Up to present, research on software quality has developed such quality models that are intended to be comprehensive and applicable to any context of software. In order to evaluate software quality, it is necessary to use a quality model which breaks software down into its different characteristics. Software engineering researchers have suggested such a various number of quality characteristics and/or criteria that they may cause confusion and not be empirically useful. Thus the ISO 9126 and are provided as a standard model for software quality and ISO 12119 is for software package. ISO 9126-1 (2001) defines six characteristics (functionality, reliability, usability, efficiency, maintainability, portability) and 27 sub-

---

<sup>1</sup> ISO/IEC 9126 is being superseded by ISO/IEC 25000 since 2005: Software engineering - Software product Quality Requirements and Evaluation

characteristics for software quality hierarchically. Each sub-characteristic is further divided into attributes.

## 2.2. Criteria Selection for Specific Software

The criteria to assess software are various according to the kind of product or evaluation objective. The organization of criteria also can follow the intention of research designer and their own scale because of comparable weights. For example of the financial transaction software, security is very important characteristic. Then security can be reorganized with other characteristics of the first level in ISO 9126 even though the security is arranged as sub-characteristic belonging to functionality in ISO 9126.

All quality models, no matter what the source is, should be tailored before being used for evaluation purpose. By tailoring the model, quality characteristics and sub-characteristics that are important and necessary to the evaluation can be focused on to include the evaluator's view point and/or application levels.

## 3. Software Quality Evaluation Model

Multiple Criteria Decision Making (MCDM) techniques have been applied to the evaluation of information systems [Chandler, 1982; Klein and Beck, 1987] and software products [Anderson, 1990; Fritz and Carter, 1994]. Among MCDM methods, weighted sum is general out of many aggregation methods. In order to get priorities or weights of quality, this study adopts the ISO 9126 quality model as evaluation criteria, and the AHP as an evaluation method. The AHP and its general form, ANP are the measurement methods for dealing with quantifiable and/or intangible criteria that has found rich application in decision theory, conflict resolution, and models of the brain because of its easy-going applicability and its ability of judgmental consistency check [Kim and Whang, 1993]. The AHP/ANP are usefully accepted when the evaluation depends on subjective judgment and/or expert knowledge. The AHP is for hierarchical decomposition with independence and the ANP is for feedback or dependent relation among criteria and/or alternatives. A characteristic advantage of the AHP/ANP is pairwise comparisons, which covers subjective and fuzzy evaluation [Saaty and Tran 2007].

The software evaluation model of this study shall be intended to make subjective judgment into objective priorities and to become a scoring model rather than a simple choice model. Matching the AHP with ISO 9126 requirements is summarized in Table 1. In brief review, the first requirement is to cover together all aspects of software quality resulting from the quality definition of satisfying implicit and explicit user's need in ISO 8402. Therefore the model is composed of hierarchic relations. The second and the third are to describe the product quality with a minimum of overlap, and to be as close as possible to the established terminology. This is not to make confusion in understanding and applying those characteristics to a software product. For instance, interoperability is used in place of compatibility in order to avoid possible ambiguity with replaceability. As a result, compatibility is broken down into interoperability and replaceability. The forth is to form a set of not more than six to eight characteristics for clarity and ease of handling.

**Table 1 Matching ISO 9126 requirements and AHP**

ISO 9126 Requirements	Related AHP Features
Hierarchic relation of quality characteristics	Hierarchic decomposed evaluation
Covering all aspects of S/W quality	Complete hierarchic structure
Description with a minimum of overlap	Criteria independent of the alternatives
Using the established terminology	Elements in a set to be compared
Forming a set of not more than six to eight	Scientific method for subjective judgment

In order to get priorities of alternatives from pairwise comparison method due to the properties of relative

and ratio scales, evaluators need to know all the given alternatives to compose the pairwise comparison matrix. However, software package evaluation model necessarily involves following usage characteristics:

- ◆ Perceived evaluation is oriented from a very limited number of products in a category,
- ◆ In case that individual assessment for a product is adopted, it is very hard to get enough number of evaluators to get a valid result for some software products.
- ◆ Paired comparison of alternatives provides more validity than single alternative evaluation.

One of the ways to raise the validity in such evaluation is pairwise comparisons. However, it is not easy to get evaluators with enough experience or knowledge to cover all the focused products. This paper proposes named pooling partial pairwise comparisons leading to full pairwise comparison matrix, which aggregates various paired comparisons of any two products from an evaluator.

One of the problems accompanied with the proposed method is to cope with missing value in the matrix. The pairwise comparison matrix fully requires  $n(n-1)/2$  times of paired comparisons. However, when paired comparisons of this paper do not lead to the fully required number of pairwise comparisons, then the missing value needs to be estimated by the other given comparison scores. Harker [1987] referred Incomplete Pairwise Comparison (IPC) to the matrix which has any missing cell and described its estimation method, in addition to Saaty's [1980]. The empirical example of this paper shows the estimation for missing values in the matrix that is composed by aggregating paired comparisons of any two products. According to the number of data, for example, different matrices can be composed. The two different vectors derived from two judgments matrices can be determined compatible or not using the compatibility index [Saaty, 2001 p 67] which is derived from the relation between compatibility and consistency.

## **4. Example**

This paper exemplifies the proposed application method to software packages. We chose five software products for PC and asked end-users to evaluate paired comparison for any of two out of five products.

### **4.1 Criteria Selection for Package and Re-organization**

Through three sets of interviews with five experts who have experienced software development for more than five years, it is found out that the practical design of a software package program is usually composed of two parts: program body and user interface. The relation matrix between the quality characteristics and the product items has been analyzed. As a result, 16 characteristics out of 27 are selected to be applied for package. The criterion to select is clear to understand from user view. First, the compliance is obviously distributed to each characteristic in ISO 9126, however this study adopts the only sub-characteristic to functionality. Second, the sub-characteristics except changeability under maintainability are deleted because maintainability is mostly applied to customized software products. Third, we re-defined here portable independence to include changeability in addition to adaptability and installability so that all factors for each level in the quality hierarchy may be equivalently sized. As a result, a new quality model for this study is restructured with 5 characteristics and 16 sub-characteristics hierarchically. Each characteristic is re-described for users to understand.

### **4.2. Priorities of Criteria**

279 users out of 455 (61.3%) responded and consistency is checked by the consistency ratio of hierarchy (CRH) (See [Saaty, 1980] p. 84). Only data of 147 respondents are selected with the condition of  $CRH < 0.2$  ( $CRH < 0.1$  would be desirable but  $CRH < 0.2$  would be tolerable [Saaty, 1980]). We try to identify how the priority or weight of each characteristic would be different along with users' skill level (beginner, intermediate and advanced) at software.

As a whole, functionality is the most importantly perceived characteristic with priority of 0.32, followed by reliability and usability in order. As hardware and OS technology has been developed immensely, so the importance of reliability might also have been increased. Next is usability and efficiency. Portable independence gets the least priority because of package traits, in that its sub-characteristics may be basically required from users' view.

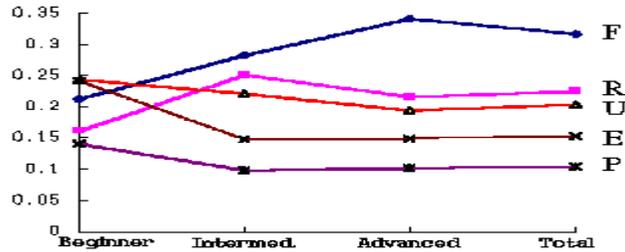


Figure 2 Priorities of Characteristics

#### 4.2. Quality Preferences of Alternatives

The aggregated quality index equals the inner product of weight vector and quality level vector for sub-characteristics. As mentioned above, this paper selected 5 products of a package to apply the proposed comparison method. We asked the package users to evaluate any possible pairwise comparison among the 5 products. Totally 126 paired comparisons are gathered. 85 respondents' data turned out to be reliable (by  $CRH \leq 0.2$  for criteria evaluation) in the evaluation of the quality hierarchy and are used for analysis. The number of respondents for each paired product appears in following Table of matrix form.

There are two missing paired comparisons which bring out IPC (incomplete pairwise comparison) matrix when we synthesize all the independent paired comparisons for each sub-characteristic. In addition, tow cells of the rest are filled with just one respondent respectively. In this paper, the two cases with 2 and 4 missing cells are analyzed and compared.

In order to get quality index from many evaluators, we take the geometric mean of users' judgments for paired products, which is suggested in the AHP, and synthesize all geometric means to construct a pairwise comparison matrix. The missing values can be estimated by the Harker's method [1987] or limiting power of the matrix by Saaty [1980]. The principal eigenvector of the complete matrix is the perceived quality preferences among alternatives. Users' preferences (weighted sum) for the 5 products are shown at the column of aggregated quality index in Table 6. However, our focus is not on the rank-parameters of the alternatives but on the empirical implementation of our model. This paper compares the two result vectors, which prove to be not different by  $SI$  (compatibility index) =  $1.005 \approx 1$ .

						Aggregated Quality Index with 4 missing (rank)	Aggregated Quality Index including * (rank)
	H1	M	H2	A	I		
H1	-	51/87	7/22	7/10	5/10	0.21 (3)	0.21 (3)
M		-	8/20	1*/2	5/9	0.23 (1)	0.24 (2)
H2			-	1*/1	0/1	0.23 (1)	0.25 (1)
A				-	0	0.20 (4)	0.18 (4)
I					-	0.13 (5)	0.12 (5)

## 5. Discussion and Concluding Remarks

Software quality is in itself one concept, however there are different perspective from the different

purpose/objective and its behavior of the computer based system that includes the software, and the effects of using the software in a specific context. Ideally, the internal quality determines the external quality and external quality determines quality in use. In real condition, it is necessary to connect quality in development with quality in use. As a result of the combination, ISO 9126-4 (efficiency, productivity, safety and satisfaction from user perspective) can be added to the model of this paper. Finally, if it is supposed that there is dependency between quality characteristics and/or alternatives when to evaluate software products, ANP should be introduced.

This paper proposed to aggregate paired comparisons used in the AHP, which can provides more validity in comparison than individual alternative evaluation can do, covering that the number of evaluators for some products is not enough and that an evaluator does not empirically experience or assess all the focused alternatives in a same category. This paper addressed to analyze two matrices each of which is respectively composed by different set of pairwise comparisons depending on the number of data from the same set of paired comparisons. Consecutively, this paper compares the results of the two matrices using the Compatibility index in the AHP, empirically. This study may become the basis for the empirical research to clarify the relation between perceived quality and objective indicators of external quality.

## **References**

- Anderson, E.E. (1990), "Choice Models for the Evaluation and Selection of Software Packages," *Journal of Management Information Systems*, 6, 123-138.
- Boehm, B.W., J.R. Brown, M. Lipow, G.J. MacLeod, and M.J. Merritt (1978), *Characteristics of Software Quality*, North-Holland, New York.
- Chandler, J.S. (1982), "A Multiple Criteria Approach for Evaluating Information Systems," *MIS Quarterly*, 6, 39-48.
- Curtis, B. (1980), "Measurements and Experimentation in Software Engineering," *IEEE Journal of Software Engineering*, 68, 1144-1157.
- Cusumano, M.A. and C.F. Kemerer (1990), "A Quantitative analysis of U.S. and Japanese Practice and Performance in Software Development," *Management Science*, 36, 1384-1406.
- Fenton, N. (1994), "Software Measurement: A Necessary Scientific Basis," *IEEE Transactions on software Engineering*, 20, 199-206.
- Fritz, C.A. and B.D. Carter (1994), *A Classification, and Summary of Software Evaluation and Selection Methodologies*, Computer Science Technical Report No. 940823, Mississippi State University.
- Harker, P.T. (1987), "Incomplete Pairwise Comparisons in the Analytic Hierarchy Process," *Mathematical Modelling*, 9, 837-848.
- IEEE 1061 (1992), *Standard for a Software Quality Metrics Methodology*.
- ISO 8402 (1994), *Quality Management and Quality Assurance - Vocabulary*.
- ISO/IEC 9126-1 (1991, 2001); 9126-2 (2003); 9126-3 (2003); 9126-4 (2004), *Software engineering - Product quality - Part 1: Quality model; Part 2: External Metrics; Part 3: Internal Metrics; Part 4: Quality in Use Metrics*.
- ISO 12119 (1994), *Information Technology - Software Packages - Quality Requirements and Testing*.
- ISO 14598-1 (1996), *Information Technology - Software Product Evaluation - Part 1 : General Overview*.
- ISO/IEC 25000 (2005), *Software Engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Guide to SQuaRE*.
- Kim S.B. and K.S. Whang (1993), "Forecasting the Capabilities of the Korean Civil Aircraft Industry," *OMEGA-International Journal of Management Science*, 21, 91-98.
- Klein, G. and P.O. Beck (1987), "A Decision Aid for Selection among Information Systems Alternatives," *MIS Quarterly*, 11, 177-186.
- Saaty, T.L. (1980), *The Analytic Hierarchy Process*, McGraw-Hill, New York.
- Saaty, T.L. (1994), "Highlights and Critical-Points in the Theory and Application of the Analytic Hierarchy Process," *European Journal of Operational Research*, 74, 426-447.
- Saaty, T.L. (2001), *The Analytic Network Process*, RWS Publications.
- Saaty, T.L. and K. Peniwati (2008), *The Group Decision Making: Drawing out and Reconciling Differences*, RWS Publications.
- Saaty, T.L. and L.T. Tran (2007), "On the Invalidity of Fuzzifying Numerical Judgments in the Analytic Hierarchy Process," *Mathematical and Computer Modelling*, 46, 962-975.