

# MEASUREMENT MODEL OF SOFTWARE QUALITY IN USER'S PERCEPTION

K. Subba Rao<sup>1</sup>, L. S. S. Reddy<sup>2</sup>

<sup>1</sup>Associate Professor, IT Department, <sup>2</sup>Professor & Director, CSE Department, LBRCE, AP, INDIA  
 ksubbarao\_22@yahoo.co.in

## Abstract

An increasing emphasis on consumer demand and expanded development budgets of software development firms fuel the need to upgrade software quality. Software quality is largely measured by quality standards and guidelines. This paper presents a method for modeling users' perception of software quality. The method aims to improve the quality of data derived from user opinion surveys and facilitate the analysis of such data. The proposed model offers a way to measure users' opinion in early stages of product release and a way of predicting the opinion subsequently formed after their opinion revisions using the initial measurements. Therefore, this work develops a conceptual software quality measurement model for evaluating software quality to decrease the perceptive and expectative (or quality) measuring gap between a software development firm and the end user's requirements.

**Index Terms:** Software quality, Software development, Quality Measurement, Quality Evaluations, & Quality Attributes

\*\*\*

## 1. INTRODUCTION

Quality is a major challenge for all complex software systems. Some important attributes of software quality include reliability, survivability, availability, safety, security, performance, accuracy, etc [1]. These have long been requirements of aerospace and defence systems. Now, equally challenging requirements are being placed on "everyday systems" that increasingly provide the infrastructure for our daily lives such as commercial, e-business and embedded systems. Assessing and controlling software quality is still an immature discipline [2][3]. One of the reasons for this is that many of the concepts and terms that are used in discussing and describing quality are overloaded with a history from manufacturing quality. Most approaches to software quality do not deal explicitly with user-perceived quality [5]. User-perceived quality is regarded as an intrinsically inaccurate judgement of product quality. Investigations seldom papers on measurement theory of software quality have been unable to provide methods in depth analyzing software quality measurement, especially in measure between the end users and the software firm [6][7]. In this paper we will discuss methods of addressing each of these problems. The key mindset is to remember that a software product is developed to provide a range of services for a user group, in order to help them achieve certain needs or goals. Thus, we should be clear at the outset of any software project as to precisely what those needs or goals are. These are the key drivers behind the identification of not just the functional requirements, but also the quality requirements. This paper presents an approach that takes business or user goals as the primary driver, and then maps these onto quality goals. Having identified a set of

quality in use goals, we next need to know how we can best control the software development life-cycle in order to be able to maximise the likelihood of achieving those goals.

## 2. SOFTWARE QUALITY ATTRIBUTES

Software Quality Attributes maps to common stakeholders' primary concerns onto quality attributes as shown below. It shows an approach to identify and resolve the conflicts among quality attributes based on product and process strategies [8]. It also shows a value-based quality model to perform trade-off analysis among conflicting quality attributes and a value based process to achieve stakeholder mutually satisfying software quality requirements [9][14].

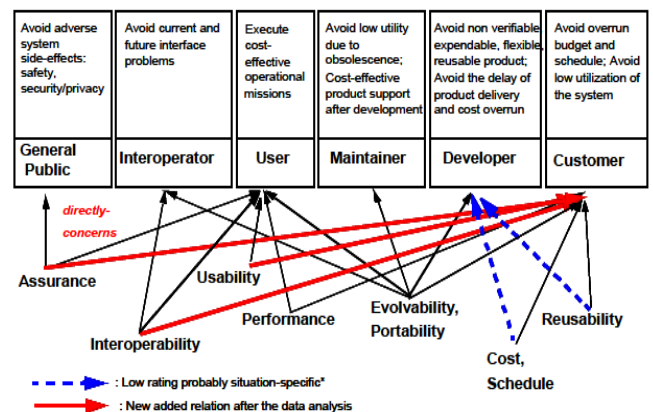


Fig: 1 First-Order Mapping of Stakeholders Primary Concerns

### 3. MODELING USERS' PERCEPTION MEASUREMENTS

In order to collect the measurements of the users' opinion, a multiple-choice format was used in the questionnaires in order to guide the user to select predefined responses that were ordered in interval scales [15][16]. Examples of the questions used in these questionnaires are the following: "What is your opinion of the product's accuracy and consistency?", "Is invalid data entry properly recognized?", "Are all functions that relate to the window available when needed?", "Is help available for each item and is it context sensitive?". QWCO (Qualifications Weighed Customer Opinion) method was used to measure users' opinions according to their qualifications.

$$QWCO = \frac{\sum_{i=1}^n (O_i \cdot E_i)}{\sum_{i=1}^n E_i} \tag{1}$$

Where  $O_i$  measures the normalised measured results of user's  $i$  opinion and  $E_i$  measures the qualifications of user  $i$ . Finally,  $n$  is the number of users who participated in the survey [11][12]. Therefore, each user contributes to the average according to his/her qualifications. Another method  $QWCO_{DS}$  (Qualifications Weighed Customer Opinion with Double Safeguards), a number of safeguards were embedded into the questionnaires [13]. Safeguards are questions placed inside the questionnaire so as to measure the correctness of responses and not aimed at measuring user perceived quality. They are control questions aiming at detecting errors.

$$QWCO_{DS} = \frac{\sum_{i=1}^n \left( O_i \cdot E_i \cdot \frac{S_i}{S_T} \cdot P_i \right)}{\sum_{i=1}^n \left( E_i \cdot \frac{S_i}{S_T} \cdot P_i \right)} \tag{2}$$

Where  $S_i$  is the number of safeguards that the user  $i$  has replied to correctly and  $S_T$  is the total number of safeguards. Since the use of the  $QWCO_{DS}$  technique implies the use at least of one safeguard in the questionnaire, division by  $S_T$  is always valid. In this method, safeguards were used not only to detect errors when measuring customer's opinion, but also to detect errors when measuring customers' qualifications [18]. In equation (2),  $P_i$  value can be 0 or 1. The value of  $P_i$  is zero when at least one error has been detected when measuring the qualifications of customer  $i$ .  $P_i$  value is set to 1 only if no error has been detected. This method results in the rejection of a customer's responses if errors were detected while measuring his/her qualifications.

### 3.1 Measuring Users' Opinion

In order to measure users' opinion of a software product efficiently, surveys in fixed time intervals must be conducted. For the analysis of the measurements, the users were divided into two main categories, the experienced and the inexperienced users. In the Fig2 the limits of the differentiation of the user's opinion over time are illustrated [10]. The horizontal bar represents the time in monthly intervals and the vertical bar represents the user's opinion, which was measured using the formulas mentioned above. The user's opinion in each survey takes values from 0 to 1. The line AvOp represents the average users' opinion of the quality of the software product, formed after the final opinion of users has been measured. The opinion of experienced users over time varies between the curves e1 and e2, whereas the opinion of inexperienced users over time varies between the curves ne1 and ne2.

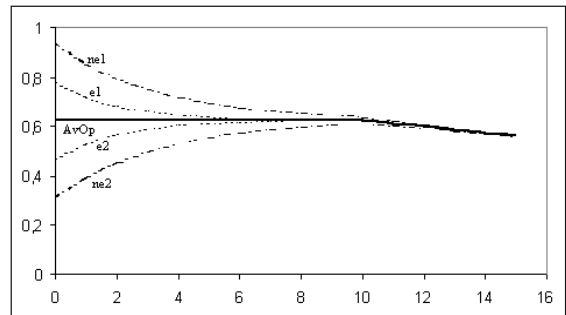


Fig: 2: Boundaries of User's opinion

The experienced users, in contrast to the inexperienced, form an opinion for the quality of the product from the early stages of its release, which is very close to their final opinion. On the contrary, the inexperienced users will form an opinion close to their final opinion after using the software product for a long period of time [9][20]. The length of this period depends on the complexity of the product, the number and the variety of the functions it supports, the amount of usage and the conditions under which usage occurs, as well as usage of similar software products. This period of time usually varies from six to twelve months, when the user is experienced in the use of this specific product [19]. After a period of time, the line AvOp usually starts to decline as the user requirements usually increase over time. This phenomenon is dependent on factors, such as the similar software products that may be released and the advances in hardware. It was also observed that when an experienced user gives the software a higher score than his final score or vice versa, this does not display fluctuations but is seemed to slowly close the gap between the high or low score and the final score. Amongst inexperienced users, however, such predictable variability was not observed; opinion fluctuated between widely ne1 και ne2. Over time, the degree of fluctuation receded to the users' final opinion of the

product quality. For example, the differentiation of inexperienced users' opinion over time can be intimated by Fig 3, where UO represents an example of the changes in a user's opinion. This fluctuation results from the inexperienced user either finding a new feature of the product, which has remained undiscovered or has uncovered some aspect of the product, which the user has sought and has not found up till now and, as a result, rates the product highly. Similarly, if the user uncovers a flaw in the product (whether real or perceived), the user will rate it lowly regardless of whether the aforementioned flaw could not have been avoided at the production stage

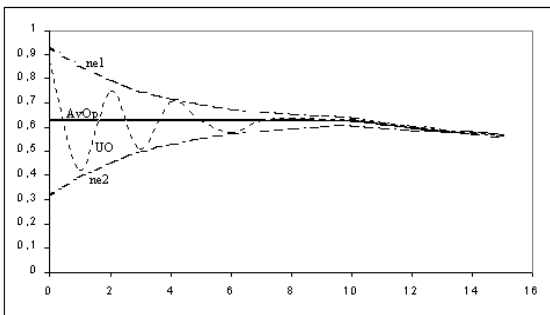


Fig. 3: Fluctuation of inexperienced users' opinion

Software quality factors are not clearly perceived by inexperienced users [7][8]. If they discover a characteristic indicating that the product fails in one particular factor, and then they consider that the product fails in all the other areas as well. On the contrary, experienced users do clearly perceive the independent nature of these factors. After a justifiable time period, inexperienced users become accustomed to the new features or flaws they discover in the product and, as a result, their opinion begin to lean towards the final opinion as is the case with experienced users[16][17].

**3.2 Findings to Improve How Surveys Are Conducted**

From the measurements of the surveys, it is obvious that over time: a) the experienced users' opinion of the quality of the software product approaches their final opinion and b) the deviation of the inexperienced users' opinion from their final opinion declines continuously. Thus, the more a customer uses a product, the more weight must be given to his opinion. In other words, the time factor must also be taken into account for effective measurements of software quality [7]. Additionally, in the early stages, the opinion of inexperienced users fluctuates greatly. Their opinion can be considered only if the sample of users is large enough to be considered representative, thus ensuring sound results. Moreover, the opinion of experienced users should be given greater weight than that of the inexperienced users, regardless of their being fewer of the former. Furthermore, from the findings for individual user groups participating in the surveys, it was also

observed that the larger the degree of fluctuation in their opinion, the more difficult it was for them to learn the features which are more relevant to their specific context [5].

**4. REPRESENTATION IN GROVE'S SYSTEM OF SPHERES**

The above representation of the constrains is sound and complete with respect to Grove's Systems of Spheres. In other words, it can be proved that each set of constrains corresponds to one and only one System of Spheres and conversely each System of Spheres corresponds to one and only one set of constrains. When designing the System of Spheres, the different possible worlds, which represent the different belief states of a user, must be placed in such a way that the belief revision rule of minimal changes during a revision will be satisfied. Thus, a system of spheres centered on [K] is a collection S of subsets of M that can be represented in Fig 4. In this system the more innermost the sphere is, the more possible the world centered on [K]. When a new sentence A appears to be true, with  $A \notin [K]$ , and A is always accepted to be reliable, the possible world must be revised in order to encompass A. So the closest sphere ( $S_A$ ) must be taken, where  $S_A \cap [A] \neq \emptyset$ , in order to have minimal changes to our first belief state. Our new world is now  $C(A)=[K*A]= S_A \cap [A]$ .

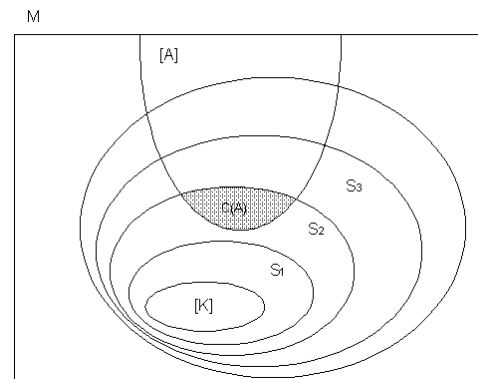


Fig. 4: Grove's System of Spheres

**4.1 Designing the Model: Premature Prediction Of Users' Opinion**

From the appropriate survey, the user's opinion for every quality characteristic will allow the researcher to plot the initial belief state of this user for the quality of the software program. Over time, this belief state may change because of the revisions of the user's opinion. The main aim of the proposed model is to determine the new belief state of the user without having to conduct another survey. The basic philosophy of this model is illustrated in Fig 4, where the inputs of the model are the current belief state of the user and

the new info of the software program that he/she has discovered and the output is his/her new belief state.

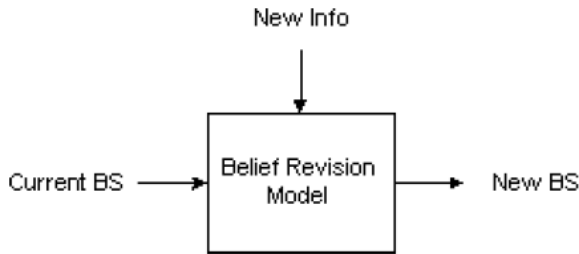


Fig. 5: The basic philosophy of the model

According to this rule, if the belief state of a user is  $\{x_1, x_2, x_3, \dots, x_n\}$ , then the closer belief states to it are all the  $\{y_1, y_2, y_3, \dots, y_n\}$ , with  $\sum |x_i - y_i| = 1$  provided that all of the above belief states are valid according to the set of constrains. These belief states will be placed into the first sphere around the  $\{x_1, x_2, x_3, \dots, x_n\}$ . Similarly, the next closer belief states are all the  $\{z_1, z_2, z_3, \dots, z_n\}$ , with  $\sum |x_i - z_i| = 2$  provided that all of the above belief states are valid according to the set of constrains. These spheres will be placed into the second sphere around the  $\{x_1, x_2, x_3, \dots, x_n\}$ . If a belief state is invalid, then it's possible world in the System of Spheres model must be placed far away from the valid ones, i.e. into the last sphere around the  $\{x_1, x_2, x_3, \dots, x_n\}$ . In this way, the above possible world will be determined as an improbable one.

4.2 Building a Set of Constrains

The representation of the correlation between the quality criteria according to users' perception of quality that was followed in the proposed model is the determination of a set of constrains between the scores of each criterion. These constrains have the following form:

$F_i C_j \in [a,b] \Rightarrow F_k C_l \in [c,d]$  which means that if the score of the criterion  $F_i C_j$  takes a value from a to b, then the score of the criterion  $F_k C_l$  will take a value from c to d. For example, the current belief state of a user is  $CBS = \{F_1 C_1 = 6, F_1 C_2 = 4, F_2 C_1 = 2\}$  and his opinion for  $F_1 C_1$  changes to 7, then this revision must have minimal changes to the CBS according to belief revision rules. As a result, according to the constrains, his new belief state will be  $NBS = \{F_1 C_1 = 7, F_1 C_2 = 5, F_2 C_1 = 3\}$ .

Table -1: Constraints Table

$F_1 C_1 \in [0,3] \Rightarrow F_1 C_2 \in [0,4]$	$F_1 C_2 \in [0,2] \Rightarrow F_1 C_1 \in [0,4]$
$F_1 C_1 \in [5,6] \Rightarrow F_1 C_2 \in [2,8]$	$F_1 C_2 \in [9,10] \Rightarrow F_1 C_1 \in [5,10]$
$F_1 C_1 \in [7,10] \Rightarrow F_1 C_2 \in [2,8]$	

$[5,10]$	
$F_1 C_1 \in [0,2] \Rightarrow F_2 C_1 \in [0,7]$	$F_2 C_1 \in [0,1] \Rightarrow F_1 C_1 \in [0,5]$
$F_1 C_1 \in [8,10] \Rightarrow F_2 C_1 \in [4,10]$	$F_2 C_1 \in [9,10] \Rightarrow F_1 C_1 \in [5,10]$
$F_1 C_2 \in [0,2] \Rightarrow F_2 C_1 \in [0,5]$	$F_2 C_1 \in [0,2] \Rightarrow F_1 C_2 \in [0,5]$
$F_1 C_2 \in [5,7] \Rightarrow F_2 C_1 \in [3,9]$	$F_2 C_1 \in [9,10] \Rightarrow F_1 C_2 \in [5,10]$
$F_1 C_2 \in [9,10] \Rightarrow F_2 C_1 \in [5,10]$	

As far as the above example concerns, the closer belief states to the belief state  $\{6,4,2\}$  are the following:  $\{5,4,2\}$ ,  $\{6,3,2\}$  and  $\{6,4,3\}$  whereas the belief states  $\{7,4,2\}$ ,  $\{6,5,2\}$  and  $\{6,4,1\}$  are invalid according to the constrains. When a revision occurs to the criterion  $F_1 C_1$  from 6 to 7, the new belief state will be the  $\{7,5,3\}$ , which is placed into the third sphere around the  $\{6,4,2\}$ . The representation of these is illustrated in Fig 6, where the current belief state  $CBS = \{6,4,2\}$ , the new belief state after the revision  $NBS = \{7,5,3\}$ ,  $A = \{5,4,2\}$ ,  $B = \{6,3,2\}$ ,  $C = \{6,4,3\}$ ,  $D = \{7,4,2\}$ ,  $E = \{6,5,2\}$ ,  $F = \{6,4,1\}$ ,  $S_i$  is the  $i^{th}$  sphere around CBS and  $K$  the boundaries of the event " $F_1 C_1 > 6$ ". The new belief state is the shaded area, which is the intersection of  $S_3$  and  $[K]$ .

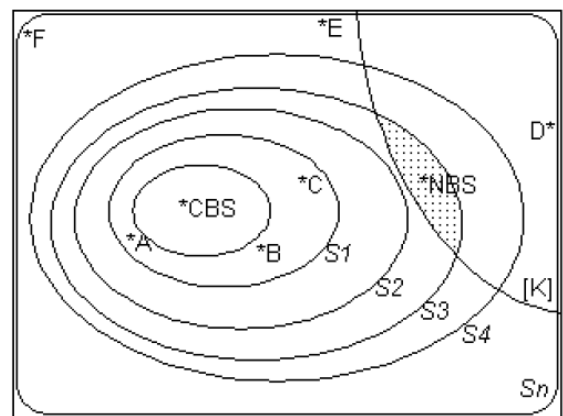


Fig. 6 Example of Grove's System of Spheres

4.3 Modeling Users Categories

The analysis revealed that users must be separated into categories according to their experience, since their opinion of a software product alters in a different way. As a result, the proposed model differs for each user category, because the revision of the opinion in one quality criterion will produce different results in the belief set of each user category. In other words, when inexperienced users discover that an event occurs or not, then their opinion for the quality of the product will change in a higher degree than the opinion of experienced

users will. Software quality factors are not clearly perceived by inexperienced users and when they are either satisfied or dissatisfied with one, the other factors follow suit. As a result, a revision in their opinion of one quality factor leads to the revision in their opinion of others. On the contrary, according to the experienced users' perception of quality, the different factors of quality are not seen as being interdependent. In the experienced user's case, a revision in the opinion of one quality factor will affect the opinion of the other factors only if this revision is of a radical nature. The different level of interdependence among software quality factors, according to users' opinion, leads to a model differently designed for each category of users. In the case of experienced users, the boundaries that declare whether an event occurs or not are presented in such a way, that a revision in an opinion of one event will result in minimal changes in their belief set. In other words, these boundaries are independent. Therefore, no areas that are dense in event boundaries are observed. Otherwise, if the belief set of a user was represented by a sphere designed into this area, a revision in the opinion of one event would lead to a radical revision, which is not observed in the case of experienced users. The model in this case can be intimated by Fig 7

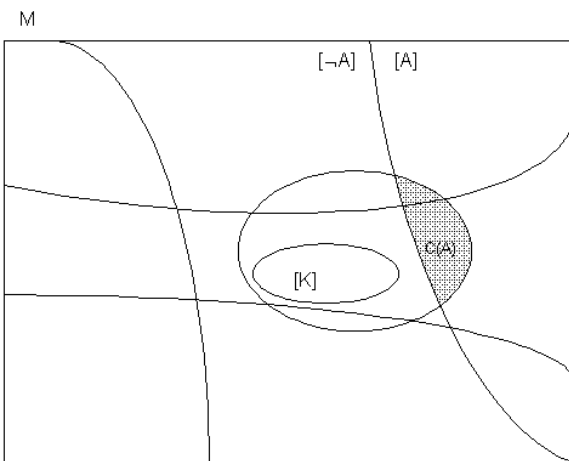


Fig: 7 Model of experienced users' category

On the contrary, in the case of inexperienced users, every new characteristic of the software product that has been detected differentiates their opinion of all the software quality factors. As a result, the model in this case is designed in such a way, that the revision in users' opinion of one factor leads to a revision of a radical nature. The possible world of inexperienced users is not as stable as in the experienced user's case. Since inexperienced users have usually the opinion that the events of the model are interrelated, the boundaries of these events must be in close proximity. The model has areas that are dense in event boundaries and it is illustrated in Fig 8

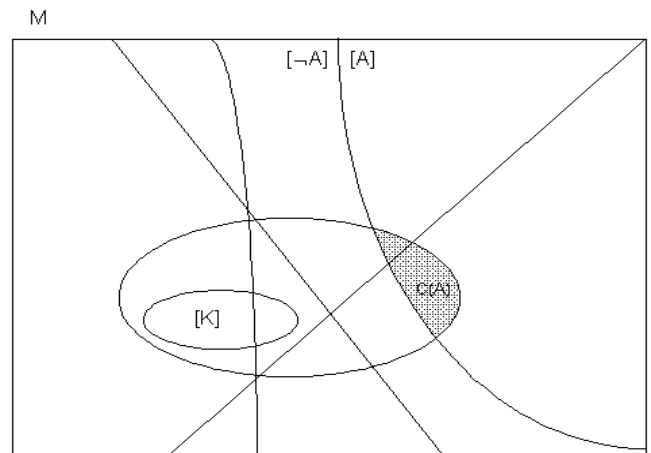


Fig 8: Model of inexperienced users' category

Figures 7 and 8 also illustrate the differentiation between the experienced users and the inexperienced users, after a belief revision. For example, in the new world  $C(A)$ , derived from the revision in the event  $A$ , inexperienced users form an opinion of all the events completely different from their initial one (opinion in world  $[K]$ ). On the contrary, experienced users could revise their opinion only in one additional event.

**CONCLUSIONS**

Using the proposed model, software production companies will be able to predict users' opinion and their revisions over time, without having to conduct surveys frequently. The boundaries of the events must be designed according to the weight given to each software quality factor. However, after a long period of time, a new survey should be conducted in order to test the soundness of this model and, occasionally, even to calibrate the way it is applied. Furthermore, software production companies will be able to predict users' opinion separately for each software quality characteristic.

**REFERENCES**

[1] Bate Roger, et al: A Systems Engineering Capability Maturity Model, Version 1.1. Software Engineering Institute, CMU/SEI-95-MM-003, November 1995.  
 [2] Brown M. G.: Baldrige Award Winning Quality: How to Interpret the Malcom Baldrige Award Criteria. Milwaukee, WI: ASQC Quality Press, 1991.  
 [3] Curtis Bill et al: People Capability Maturity Model. Software Engineering Institute, CMU/SEI-95-MM-02, September 1995.  
 [4] Cai, Shaohan & Jun, Minjoon. (2003). Internet users' perceptions of online service quality: a comparison of online buyers and information searchers. *Managing Service Quality*, 13(6), 504 – 519

- [5] Carlson, Jamie & O'Cass, Aron. (2010). Exploring the relationships between e-service quality, satisfaction, attitudes and behaviours in content-driven e-service web sites. *Journal of Services Marketing*, 24(2), pp.112 – 127
- [6] Dabholkar, P.A. (1996), Consumer evaluations of new technology-based self-service options: an investigation of alternative models of service quality. *International Journal of Research in Marketing*, 13, 29-51.
- [7] Evanschitzky, H., Iyer, G., Hesse, J. and Ahlert, D. (2004). E-satisfaction: a re-examination. *Journal of Retailing*, 80(3), 239-47
- [8] Gounaris, S., Dimitriadis, S. and Stathakopoulos, V. (2005). Antecedents of perceived quality in the context of Internet retail stores. *Journal of Marketing Management*, 21(7), 669-682.
- [9] Gardenfors Peter: *Knowledge in Flux – Modeling the Dynamics of Epistemic States*. MIT Press, Cambridge, England, ISBN 0-262-07109-6, 1988
- [10] Grove Adam: *Two modelings for theory change*. *Journal of Philosophical Logic*, 17, 157-170, 1988
- [8] Peppas Pavlos: *Well Behaved and Multiple Belief Revision*. *European Conference on Artificial Intelligence*, 1996
- [9] Heim, G.R. and Field, J.M. (2007). Process drivers of e-service quality: analysis of data from an online rating site. *Journal of Operations Management*, 25, 962-84
- [10] Kassim, N. and N. Abdullah. (2010). The effect of perceived service quality dimensions on customer satisfaction, trust, and loyalty in e-commerce settings. *Asia Pacific Journal of Marketing and Logistics*, 22(3), 351-371.
- [11] Kim, M. and Stoel, L. (2004). Apparel retailers: Web site quality dimensions and satisfaction. *Journal of Retailing and Consumer Services*, 11, 10-91
- [12] Li, H. and Suomi, R. (2009). A proposed scale for measuring e-service quality. *International Journal of u- and e-service, Science and Technology*, 2(1), 1-10.
- [13] Loonam, Mary & O'Loughlin, Deirdre. (2008). Exploring e-service quality: a study of Irish online banking. *Marketing Intelligence & Planning*. 26(7), 759 – 780.
- [14] Paulk M., Curtis B., Chrissis M. and Weber C.: *Capability Maturity Model for Software*, Software Engineering Institute, CMU/SEI-93-TR-024, 1993
- [15] Ruyter, K.D., Wetzels, M. and Kleijnen, M. (2001). Customer adoption of e-service: an experimental study. *International Journal of Service Industry Management*, 12(2), 184-207
- [16] Xenos M. and Christodoulakis D.: *Software Quality: The User's Point of View*. pp. 266-272 of *Software Quality and Productivity*, Chapman & Hall, ISBN: 0-412-62960-7, 1995.
- [17] Xenos M., Stavrinoudis D. and Christodoulakis D.: *The Correlation Between Developer-oriented and User-oriented Software Quality Measurements (A Case Study)*. 5th European Conference on Software Quality, EOQ-SC, Dublin, pp. 267-275, 1996
- [18] Xenos M. and Christodoulakis D.: *Measuring Perceived Software Quality*. *Information and Software Technology Journal*, Butterworth Publications UK, Vol. 39, pp. 417-424, June 1997.
- [19] Yang, Z. (2001). Consumer perceptions of service quality in Internet based electronic commerce. *Proceedings of the EMAC Conference*, 811 May 2001, Bergen.
- [20] Yen, C. and Lu, H. (2008). Effects of e-service quality on loyalty intention: an empirical study in online auction. *Managing Service Quality*, 18(2), 127-146