

FQWCOS: A Flexible Model for Measuring Customer Satisfaction on Software Based Products and Service

Ezekiel Uzor Okike

Department of Computer Science, University of Botswana, Gaborone, Botswana

Email address:

euokike@gmail.com

To cite this article:

Ezekiel Uzor Okike. FQWCOS: A Flexible Model for Measuring Customer Satisfaction on Software Based Products and Service. *Software Engineering*. Vol. 6, No. 4, 2018, pp. 110-115. doi: 10.11648/j.se.20180604.11

Received: November 10, 2018; **Accepted:** December 11, 2018; **Published:** December 11, 2018

Abstract: Many software products and services deployed in user environments at times fail to meet user needs satisfactorily. This may be due to the fact that the product or service failed to meet user requirements from the outset (inception) of the Information Systems (IS) project. This study proposes a Flexible Qualifier Weighted Customer Opinion with Safeguard Estimates (FQWCOS) model for measuring the satisfaction of users of software products and services. The FQWCOS model is a variant of the Qualifications Weighted Customer Opinion with Safeguard questions (QWCOS). The FQWCOS model was verified with empirical data using samples from 40 users of ASAS software product. Descriptive statistics were also used to obtain the frequencies, mean values, relative frequencies, standard error, and standard deviation. From these values, it was possible to compute the normalized score of customer opinion O_i and the external measures E for QWCOS and E_i ($i=1-4$) for FQWCOS were computed. Results from the study reveal that there was no difference between the external measures for QWCOS and FQWCOS. However, the result suggest that external measures were higher when standard error (SE) was used to obtain the measures at different levels 31.58, 19.79, 21.76, 35.69 and 31.06 than when external measure was computed using standard deviation (STD) which yielded the values 4.99, 3.13, 3.44, 5.64 and 4.07. We conclude that FQWCOS and QWCOS yield the same values probably due to small sample used. However, FQWCOS provides a flexible and simple approach, and reveals the need to use the standard error instead of standard deviation since this yields higher magnitude values appropriate for expressing external measures in percentages.

Keywords: Software Quality, External Measurement, Customer Satisfaction, Flexible Model

1. Introduction

Software metrics presently preferred to be called software measurement is concerned with measurements in Software Engineering. Software Engineering (SE) is concerned with all the activities in software production from specification to maintenance, including issues of project management, use of tools, methods and theories as applicable [27].

The preference of the term “Software measurement” to the term “Software metric” is due that the fact that the term “measure” supposedly emphasizes that the collection of measurements preserve relationships that exist between the entities being measured. This is the approach of the science of measurement otherwise called measurement theory [8, 13].

Software applications today are becoming more complex,

and organizations cannot afford to neglect the likely consequences of the failure of their Information Systems (IS). Among these consequences are economic losses, and even threats to human lives. Therefore, a means of effectively measuring the quality and reliability of software products is needed. On the other hand, effective use of measurement instruments in software products and services benefit an organization in many ways [5]:

1. Estimating the cost and schedule of future projects
2. Evaluating the productivity impacts of new tools and techniques
3. Establishing productivity trends over time
4. Improving software quality
5. Forecasting future staffing needs
6. Anticipating and reducing future maintenance requirements
7. Setting design standards for an organization

1.1. Statement of the Problem

The need to continuously measure customer satisfaction as a dominant quality in use factor for software products and services cannot be over emphasized [10, 18]. Although there are models for measuring customer satisfaction [1, 3, 14], their approaches have shortcomings and may not be appropriate to software products and services. For instance, the Qualifications Weighted Customer Opinion with safeguard questions (QWCOS) [3] is statistically good but does not allow for flexible customer qualifiers, which should have been indeed quantifiers. Therefore, it becomes very imperative to evolve a model for measuring customer satisfaction which allows for flexible quantifiers.

1.2. Aims and Objectives

The aim of this study is to propose a model for measuring satisfaction as a quality indicator in software products and services, hence, measure the effect of customer satisfaction in delivered software products or services using empirical data.

The rest of this paper counts of four sections. Section 2 presents a review of related work. Section 3 presents the empirical study, methodology, result and discussion. Section 4 presents the conclusion.

2. Literature Review

2.1. Software Quality Measurement

Measuring quality in software products or service may be done in two ways namely [21]:

1. Measuring internal attributes of software at code level.
2. Measuring external attributes of software in a user environment.

This study recognizes a third component of software quality measurement as:

3. Measuring Quality in the context of use as suggested in the ISO/IEC quality measurement structure models 25022 – 25024 (2015) [10]

A software quality model that represents these approaches is evident in figure 1.

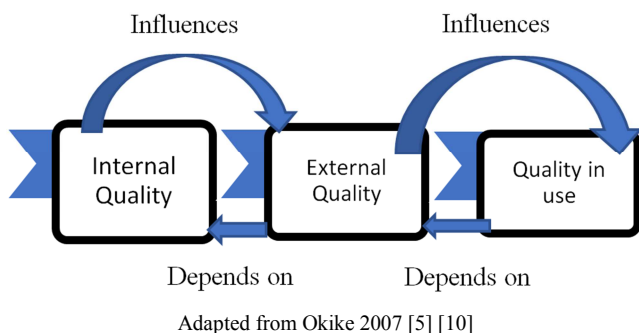


Figure 1. Software quality measurement model.

From figure 1 internal software quality influences the external software quality, and the external software quality influences the quality in use. Further explanations about

software quality attributes and models are provided in the following subsections

2.1.1. Internal Software Attributes

(Code Level Attributes Measurement).

Measuring Software Product Quality from Code level perspectives usually takes into account the internal software attributes which make up the structural complexity of a software program. Software complexity has multiple facets, including algorithmic complexity and structural complexity [5]. The structural complexity of a program has been defined as *the organization of program elements within a program* [23]. Measures of structural complexity are internal attributes of software, that is, they are specific to the software artifacts. Figure 2 identifies seven sources of software complexity namely: software, module cohesion, module coupling, control structures, algorithm, data structures, and nesting level (ibid). These are the recognizable and measurable internal software quality attributes. Several metrics and models exist for measuring software quality from code level using cohesion and coupling metrics [4, 7-9, 11-13, 15-17, 19, 20, 22, 28-30].

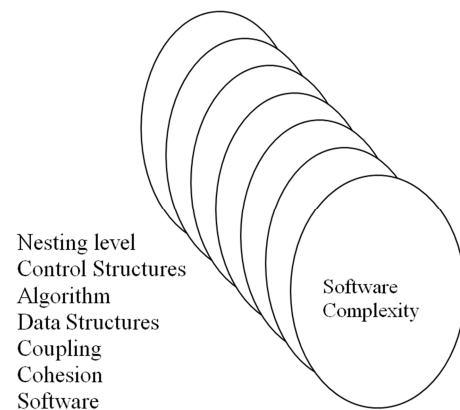


Figure 2. Different levels of software complexity.

2.1.2. External Software Attribute Measurement

External software quality attributes are those attributes of the software observable from use in the user environment as opposed to attributes inherent in the software codes. External product attributes depend on both the product behavior and the environment. Examples of external software attributes are usability, integrity, efficiency, testability, reusability, portability, interoperability, understandability, maintainability, reliability e.t.c. as shown in figure 3. Most external software attributes are non functional system requirements (sometimes called quality of service requirements). These include:

1. Performance. Example speed, process transaction time, user response time, screen refresh time.
2. Usability. The ease of use of a system (software)
3. Scalability. The ability of a system to adapt to increased demand
4. Security. System protections from unauthorized accesses to computer files and other forms of attacks on the system
5. Reliability. The meantime to failure, rate of failure occurrence, probability of unavailability, availability of

a system

6. Robustness in terms of time to restart after failure, probability of data corruption on failure etc

There exist various metrics for measuring each aspect of h software systems external attributes from software engineering perspectives [22].



Figure 3. External Software Quality attributes.

As illustrated in Figure 1, internal quality attributes influence external quality attributes which in turn influence quality in use. On the other hand, quality in use depend on external quality attributes, which in turn depend on internal quality attributes.

2.1.3. Measuring Quality in use Attribute

The quality in use of a software product is defined as perceptions of the users of the software in the context of the user environment. The measureable attributes as identified in the ISO/IEC document include effectiveness, productivity, safety, and satisfaction [10]. An adapted version of this model is shown in figure 4.



Figure 4. Quality in use

This study is concerned with measuring user satisfaction based on the external quality in use attribute of satisfaction, from the users' perception of the software in the user environment [6, 10].

2.2. Measuring Customer Satisfaction

Customer (User) Satisfaction in the context of this paper means the act of satisfying the needs of the user of a software product or service. Hence, satisfying a user implies fulfilling the desires or needs of the user with respect to the product or service. In general the business domain views customer satisfaction as the core of human experience which reflects *our liking of a company's business activities* (or products). Therefore, *a high level of customer satisfaction are strong predictors of customer retention, customer loyalty, and product or (service) repurchase* [1, 14, 18, 26].

Providing customer satisfaction could be a quality criteria or requirement for software products or services. In this case, customer satisfaction means meeting all the customer requirements. According to Asher [18], customer satisfaction for a product or service may be measured by answering the following questions:

1. Did the provider keep all promise about the product or service to the user?
2. Did the user get good service from the product or service?
3. Did the provider do everything to meet the user needs?
4. Was the product or service delivered on schedule?
5. Did the provider start for the users' problem rather than the product or service?
6. Are the providers aware of the all the user needs, current and potential? Did they ask the users for their needs?
7. Did the provider notify the user of any changes to the product or service?
8. How many providers team members understand the needs of the users?
9. Can the users depend on the quality of the product or service?
10. Are providers interested in how users use the product or service?

From the forgoing, it is clear that the end product of every service rendered, or product usage is customer (user) satisfaction. Customer satisfaction is thus a form of evaluation of a product, in this case software. Such an evaluation, which among other factors measures user's perceptions of a product, could be useful in generating information that could eventually lead to improvement of the product. To measure customer's satisfaction, this paper proposes a customer satisfaction model, which takes into account ISO quality models as shown in figures 2-4 above. We demonstrate the usefulness of the proposed model with an empirical study of customer satisfaction on a software product for academic management and administration of students' grades (ASAS) which is used in many Southern African universities and especially at the University of Botswana.

2.3. Academic and Students Administration System (ASAS)

The Academic Student Administration System (ASAS) is a generic software product. Generic products are sold on the open market to any customer who is able to buy the product, as opposed to bespoke (or customized) software products which are commissioned and developed for a particular customer or user. With generic products such as the ASAS software tool, purchasing organizations may sometimes reconfigure the software to suite the environment where it must be used. However, one peculiar challenge with generic products that require further configurations before use is that the IT personnel usually would do the configuration based on their technical knowledge without having to involve the users for appropriate user requirements input. This approach should create a knowledge gap between requirements analysis and software design as usual [2, 25]. Therefore, there is often the need for measuring user (customer) satisfaction on generic products such as the ASAS system to ascertain their usefulness to users or otherwise in the environment from time to time.

3. The Empirical Study: Modeling and Measuring Customer Satisfaction on ASAS Software

3.1. The Method

The study employs two models for measuring customer satisfaction namely the Qualifications Weighted Customer Opinion with Safeguards (QWCOS) model and a Flexible Qualifier Weighted Customer Opinion with Safeguards (FQWCOS) model proposed in this study. *QWCOS techniques estimate the result of the external measurement (EM), which is based on O_i (the normalized score of customer opinion), the E_i (qualifications of customer i) and the use of a number of control questions (safeguard questions S where S_T is the total number of control questions, from which the customer i (from a total number of n customers) has responded correctly as at S_i [3]:*

$$EM = QWCOS = \sum_{i=1}^n \left(O_i \cdot E_i \cdot \frac{S_i}{S_T} \right) / \sum_{i=1}^n \left(E_i \cdot \frac{S_i}{S_T} \right) \quad (1)$$

In this study, we define a flexible external measure M_E having a flexible Qualifier Weighted Customer Opinion with Safeguard estimates (FQWCOS).

Formally, define External Measure (M_E) as a quadruple ($\Sigma, O_i, \alpha, \beta$). Hence,

$$M_E = \sum_{i=1}^n (O_i \cdot \alpha_j \cdot \beta) / \sum_{i=1}^n (\alpha_j \cdot \beta) \quad (2)$$

Where:

O_i = normalized score of user i opinion using mean Standard Error (S.E) or Standard Deviation (STD)
 α = Flexible qualifier of user
 β = relative frequency
Hence $M_E = O_i$ (best measured using S.E).

From Eq.(2), external measure M_E may be applied in any number of steps as desired by varying the flexible qualifier α_j ($\alpha_j, j=1- n$). In this study, quantity (α_j is applied in four steps as shown in equations (3) to (6):

Step 1 (using the highest academic qualifications of users as in Eq 1)

$$M_{E1} = FQWCOS = \sum_{i=1}^n (O_i \cdot \alpha_1 \cdot \beta) / \sum_{i=1}^n (\alpha_1 \cdot \beta) \quad (3)$$

Step 2 (using IT technical competence of users)

$$M_{E2} = FQWCOS = \sum_{i=1}^n (O_i \cdot \alpha_2 \cdot \beta) / \sum_{i=1}^n (\alpha_2 \cdot \beta) \quad (4)$$

Step 3 (using years of experience in using the product)

$$M_{E3} = FQWCOS = \sum_{i=1}^n (O_i \cdot \alpha_3 \cdot \beta) / \sum_{i=1}^n (\alpha_3 \cdot \beta) \quad (5)$$

Step 4 (using users over all satisfaction with the product)

$$M_{E4} = FQWCOS = \sum_{i=1}^n (O_i \cdot \alpha_4 \cdot \beta) / \sum_{i=1}^n (\alpha_4 \cdot \beta) \quad (6)$$

The essence of the four steps is to observe the effect of variable α_j in term of the over all magnitude of quantity M_E as the highest measure of customer satisfaction of the product or service in a user environment. Both the QWCOS and the FQWCOS models were verified with data collected from 40 users of ASAS software product using a questionnaire instrument. Results of both models were comparatively analyzed to determine which of the models best measures customer satisfaction based on some threshold.

3.2. Result and Discussion

The results of measuring user satisfaction using the QWCOS and the FQWCOS models are shown in tables 1 and 2. From Table 1, using 40 respondents and the safe guard questions S where S_T is the total number of control questions, from which customer i , (from a total number of n customers) has responded correctly as at S_i [2], the value of O_i was computed using either the mean standard error or the standard deviation. The magnitude of O_i using mean standard error (S.E) is higher than O_i using the standard deviation. Furthermore, from Table 2, the external measures for QWCOS and FQWCOS remain the same. Hence, QWCOS and FQWCOS are comparatively the same.

Table 1. Measuring users satisfaction with QWCOS and FQWCOS.

QUESTIONS	N = 40	FREQ. 4-5 1-3	MEAN (μ)	REL. FREQ	S.E / STD	$O_i = \frac{\mu - \mu}{S.E}$	$O_i = \frac{\mu - \mu}{Std}$
1 Level of satisfaction with ASAS	40	6 34	2.6000	1.00	.17097 1.08131	31.58	4.99
2 Pretty Satisfied with ASAS	40	19 21	2.4250	1.000	.28168 1.7849	19.79	3.13
3 Pretty certain I can Use ASAS without assistance	40	14 26	2.2750	1.00	.26309 1.66391	21.76	3.44
4 I have truly enjoyed using ASAS	40	7 33	2.7000	1.000	.14850 .93918	35.69	5.64
5 I can recommend ASAS to other institutions	40	15 25	3.2308	1.000	.15351 .95866	31.06	4.97
6 Experience in years in ASAS	40	30 10	3.9250	1.00	.15766 .99711	25.85	4.09
7 IT competence	40	11 29	1.4750	1.000	.12904 .81610	50.57	8.00

Table 2. Comparative analysis between QWCOS and FQWCOS.

QUESTIONS	QWCOS	FQWCOS	$O_i = \frac{\mu - m}{S.E}$	$O_i = \frac{\mu - m}{Std}$
1 Level of satisfaction with ASAS	EM	$M_{E1} / M_{E2} / M_{E3} / M_{E4}$	31.58	4.99
2 Pretty Satisfied with ASAS	EM	$M_{E1} / M_{E2} / M_{E3} / M_{E4}$	19.79	3.13
3 Pretty certain I can Use ASAS without assistance	EM	$M_{E1} / M_{E2} / M_{E3} / M_{E4}$	21.76	3.44
4 I have truly enjoyed using ASAS	EM	$M_{E1} / M_{E2} / M_{E3} / M_{E4}$	35.69	5.64
5 I can recommend ASAS to other institutions	EM	$M_{E1} / M_{E2} / M_{E3} / M_{E4}$	31.06	4.97
6 Experience in years in ASAS	EM	$M_{E1} / M_{E2} / M_{E3} / M_{E4}$	25.85	4.09
7 IT competence	EM	$M_{E1} / M_{E2} / M_{E3} / M_{E4}$	50.57	8.00

4. Conclusion

Both models QWCOS and FQWCOS are comparatively the same. However, FQWCOS computation appear more straight forward and also flexible with the explanation provided in this study. FQWCOS approach suggest that the impact of the external measure of satisfaction on the selected software product yields higher values when the normalized score of customer opinion O_i is computed using the mean standard error (S.E) than using the standard deviation as shown in table 2.

References

- [1] D. A. Baker and J. L. Crompton, "Quality, satisfaction and behavioral intentions", *Annals of Tourism research*, Vol. 27, no. 3, pp. 785-804, 2000.
- [2] D. Dungan. "Enterprise Software Architecture and Design: Entities, Services, and Resources". USA: Wiley, 2012, pp. xxiii+481.
- [3] D. Stravrinoudis, and M. Xenos, "Comparing internal and external software quality measurements," *Proceedings of the 8th Joint Conference on Knowledge-Based Software Engineering*, Piraeus Greece: IOS Press, August 25-28, pp. 115-124, 2008.
- [4] E. J. Weyuker, "Evaluating software complexity measures," *IEEE Transactions on Software Engineering*, vol. 14, no. 9, pp. 1357– 1365, 1988.
- [5] E. U. Okike, "Measuring Class Cohesion in Object-Oriented Systems Using Chidamber and Kemerer Metrics and Java as Case study," P.HD Thesis, Department of Computer Science, University of Ibadan, xvii+133 pp, 2007.
- [6] E. U. Okike, and Merapelo Mogorosi, "Measuring the usability probability of learning management software using logistic regression model", *Computing conference*, London, UK, 18-20 July, 2017.
- [7] E. U. Okike, "A proposal of Normalized Lack of Cohesion in Method (LCOM) metric using field experiment", *International journal of Computer Science Issues*, vol. 7, issue 4, no. 5, pp. 19-27, 2010.
- [8] E. Okike and A. Osofisan, "A validation of Chidamber and Kemerer's LCOM metric using measurement theory", *Journal of Applied Information Science and Technology*, vol 1 issue 1, pp. 30-43, 2007.
- [9] F. Beck, and S. Dichl, "On the congruence of modularity and code coupling", *Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering*, ACM, NYUSA, 354-364, 2011
- [10] ISO 25022-250240 Quality Measurement structure and revised models, 2015
- [11] J. A. Dallal, "Measuring the discriminating power of Object-oriented class cohesion metrics", *IEEE transactions on software engineering*, 37 (6), 788-804, 2011.
- [12] J. M. Bieman and B. K. Kang, "Measuring design level cohesion", *IEEE transactions on Software Engineering*, 20, 2: 111-124, 1998.
- [13] J. Michura, M. Capretz and S. Wang, "Extension of Object-oriented metrics suite for software maintenance", *Hindawi Publishing Corporation ISRN Software Engineering Volume 2013, Article ID 276105, 14 pages, 2013* <http://dx.doi.org/10.1155/2013/276105>

- [14] H. Oh, "Service quality, customer satisfaction, and customer value: A holistic perspective", *Hospitality Management* 18, 67-82, 1999.
- [15] H. S. Chae, Y. R. Kwon, and D. H. Bae, "A cohesion measure for classes in object-oriented classes," *Software*, vol. 30, no. 12, pp. 1405–1431, 2000.
- [16] L. Badri, and M. Badri, "A proposal of a new class cohesion criterion: an empirical study", *Journal of object technology*, 3, 4: 145-159, 2004.
- [17] L. Etzkorn, J. Bansiya, and C. Davis, "Design and code complexity metrics for OO classes," *Journal of Object-Oriented Programming*, vol. 12, no. 1, pp. 35–40, 1999.
- [18] M. Asher. Measuring customer satisfaction, *The TQM magazine*. Vol. Iss. 2. Pp 93-97. 1989.
- [19] M. Bundschuh and C. Dekkers, "Object-oriented metrics," in *The IT Measurement Compendium*, M. Bundschuh and C. Dekkers, Eds., pp. 241–255, Springer, Berlin, Germany, 2008.
- [20] M. H. Tang, M. H. Kao, and M. H. Chen, "An empirical study on object-oriented metrics," in *Proceedings of the 6th International Software Metrics Symposium*, pp. 242–249, November 1999.
- [21] N. E. Fenton, and S. L. Pfleedger, "Software Metrics: A rigorous & Practical Approach," 2nd ed., London: PWS, pp. xii+638, 1997.
- [22] N. F. Schneidewind, "Systems and Software Engineering with Applications", USA: Standards Information Network, IEEE press, xix+ 443pp, 2009.
- [23] N. Gorla, and R. Ramakrishnan, "The Effect of Software Structure attributes on Software Development Productivity," *Journal of Systems and Software*, 36, 2: 191-199, 1997.
- [24] P. D. Dacrcy; C. F. Kemerer; S. A. Slaughter, and J. E. Tomayko, "The structural complexity of software: an experimental test", *IEEE transactions on Software Engineering*, 32. 1: 54-64, 2005.
- [25] P. Ralph. "The illusion of Requirements in Software Development," *Requirements Engineering* (2013) 18, pp. 293-296.
- [26] S. M. Smith, "Customer Satisfaction Survey Questions: 5 Sample Templates you can use right away," [www. Qualtrics.com](http://www.Qualtrics.com) retrieved 26 July, 2018.
- [27] S. Ian, "Software Engineering", 9th ed., USA: Pearson, 2011, pp. xiii+773.
- [28] S. Sarkar, A. C. Kak, and G. M. Rama, "Metrics for measuring the quality of modularization of large-scale object-oriented software," *IEEE Transactions on Software Engineering*, vol. 34, no. 5, pp. 700–720, 2008.
- [29] V. R. Basili, L. C. Briand, and W. L. Melo, "A validation of object-oriented design metrics as quality indicators," *IEEE Transactions on Software Engineering*, vol. 22, no. 10, pp. 751– 761, 1996.
- [30] Y. Zhou, B. Xu, and H. Leung, "On the ability of complexity metrics to predict fault-prone classes in object-oriented systems," *Journal of Systems and Software*, vol. 83, no. 4, pp. 660–674, 2010.