

A Quality Factor for Software

Luigi Buglione

European Software Institute
Parque Tecnológico de Zamudio #204
E-48170 Zamudio, Bizkaia, Spain
E-mail: Luigi.Buglione@Esi.es
Buglione@tin.it,
Tel: (34) 94 420 9519
Fax: (34) 94 420 9420

Alain Abran

Software Engineering Management Research Laboratory
Université du Québec à Montréal
C.P. 8888, Succ. Centre-Ville
Montréal, Québec, Canada
E-mail: abran.alain@uqam.ca
Tel: (514) 987-3000 (8900)
Fax: (514) 987-8477

Summary – *This work starts from the analysis of the increasing importance for management to have available tools for quality measurement of company resources, in particular of software. This work presents the concept of the design of a Quality Factor (QF) for a qualitative software evaluation, considering three distinctive but connected areas of interest, each of them representing dimension of performance:*

- ***economic dimension**, the perspective is the managers' viewpoint;*
- ***social dimension**, the perspective is the users' viewpoint;*
- ***technical dimension**, the perspective is the developers' viewpoint.*

An implementation of this QF, based on ISO/IEC 9126 standard, in quality models is also discussed.

Key words – Software Quality, Quality Factor, ISO/IEC 9126, Quality Models.

1. Introduction

Measurement and assessment, both of products and processes, are becoming a most important topics in the software engineering community. They are being increasingly recognized as being fundamental to objectively assess and to set realistic targets to improve organizational performance, with a view both to resources allocation and functional process areas implementation – or improvement, in order to reach optimal qualitative levels (1).

Measuring its own projects performance levels becomes then a strategic component for a proper planning and development of the software organization.

But in order to do that, in the evaluation you must consider the viewpoints of multiple company *dimensions* (6), all part of the software production process, such as:

- the **Economic** one, represented by the managers' viewpoint;
- the **Social** one, represented by the users' viewpoint;
- the **Technical** one, represented by the developers' viewpoint.

Nearly the totality of software engineering literature takes into consideration only the first and the third viewpoints cited above. Because of a growing involvement of users with information technologies, the second viewpoint must now play an essential role in software assessment, because they are the real users of these products; it is increasingly necessary to meet their software quality requirements and only by considering the three dimensions at the same time is it possible to obtain correct and complete assessments.

Referring to product quality, specifically of software, it must be interpreted in the light of the concept of **purpose of use**, considering both internal attributes (product characteristics) and the external ones (aim of the use). From the different possible definitions of quality, we refer here to “the totality of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs” (12). Therefore, assessments of software quality should take into account multiple and distinct viewpoints such as the three ones discussed above:

- the **management's** one, who is “interested in the overall quality rather than in a specific quality characteristic [...] and need to balance the quality improvement with management criteria” (13);
- the **user's** one, whereas software quality is given by all the properties to satisfy correctly and efficiently the present and future real needs of who buys and uses it;
- the **developer's** one, whereas software quality is given by the “conformity to functional and performance requirements explicitly fixed, to development standards explicitly documented and to implied characteristics expected of every software developed in a professional way” (20).

This paper presents a Quality Factor (**QF**) for software to give quality a value, relevant for all three viewpoints considered. QF uses an *open* weight scale methodology, to let you use the preferred measurement scale. Since the fundamental relevance to permit the adoption of a common IT language and to have an external comparability of your own internal results, ISO/IEC 9126 standard (12) was chosen as a basis to build the tools for QF calculation.

2. Quality Models

A Quality Model (**QM**) is defined as “the set of characteristics and the relationships between them which provide the basis for specifying quality requirements and evaluating quality” (12) or as a “structured set of properties required for an object of a class to meet defined purposes” (10). The goal of this paper is, as already said, to present a quality value, expressed by QF. QF represents the output of a qualitative evaluation process based on a QM.

The benefit of QMs is given by the decomposition of the valuable object (both a process, a product or an organization) in a list of its characteristics, subcharacteristics and measures and it is applicable both to predict/assure/verify the achievement of a defined goal about the object before/while/after producing it.

The best known QMs for software are those proposed by Boehm et al. (5) and McCall et al. (18), from which is originated the current ISO/IEC 9126 standard^{1 2}.

¹ The current standard, released in 1991, is composed by 6 characteristics and 21 subcharacteristics.

² We have not listed IEEE Std 1061-1992 (11) in the text because it does not propose a pre-compiled checklist, but just furnishes in Annex A an example list of factors, subfactor and metrics specifying that “these tables are not exhaustive. There are many possible sets of factors, subfactors and metrics that can be used”. By the way, this standard lists the same six ISO characteristics and 18 subcharacteristics with a strong overlapping with the 21 ones in the ISO model.

It is possible to distinguish QMs depending on their number of layers:

- **2 layers** (Boehm and McCall), that show a list of characteristics subdivided in a set of subcharacteristics; up to two levels we have a description closer to user's viewpoint;
- **3 layers**, where it is added a detailed level of single subcharacteristic measures, of course closer to a technical perspective³. It allows to quantify an evaluation in a extremely subjective manner (subjectiveness is intrinsic in QMs), equally important as the objective one (19).

For clarity of definition, we use in this study the ISO terminology. Table 1 maps the different terms used in these various software QMs:

LAYER	BOEHM (4)	MCCALL (18)	ISO (13)	IEEE (11)	DROMEY (9)
1	H-Level Charact.	Factor	Characteristic	Factor	H-Level Attribute
2	Primitive Charact.	Criteria	Subcharacteristic	Subfactor	Subordinate Attribute
3 ⁴	(Metric)	(Metric)	(Metric)	Metric	

Table 1- *Software Quality Models*

Another classification of QMs is about the number of relationships between the first two layers:

- **1:n relationship**, as in ISO 9126 - every characteristic has its own set of subcharacteristics;
- **n:m relationship**, as in McCall Factor-Criteria Model (FCM) - every subcharacteristic is linked to one or more characteristics.

However it is argued in (9) that with these models:

- it is not sufficient to list some characteristics that imply or contribute to a better quality in building software, but it is necessary to create direct links between quality attributes and product characteristics;
- there is a little assistance in building quality into software.

The problem of a common acceptance of more relevant characteristics to consider or to reject is a “never-ending story” in every community and also in the IT one. Therefore the value of standards is to group as many people as possible at the aim to use and share definitions, concepts and methods.

For these reasons we did not defined our own QM but selected the ISO quality model. We also took into account the current ISO work-in-progress improving of the standard (14, 15, 16), to have a mapping of current and future lists of characteristics for comparable QF values and formulas.

3. Quality Factor: the procedure flow

This section presents the design of the procedure to calculate QF, which returns a numerical value integrating the users' (U), developers' (D) and managers' (M) opinions about the quality of the software being measured. As mentioned above, it is important to a company to do regular quality assessments of its products in order to manage them properly. Management must therefore obtain a value to decide, based on the opinions of the three major interest groups (U, D, M), which actions must be taken for the future.

Figure 1 shows the high-level procedure flow. First, a questionnaire is submitted to both users and developers and managers, who express their quality opinions. Then, the questionnaires are gathered and consolidated these opinions and, through the QF calculation procedure, the final quality value is obtained for the software project being assessed. The procedure is fairly simple and easy to automate with a spreadsheet.

³ It is possible to think to this classification as comparable to GQM paradigm by Basili & Rombach (4), where Goals and Questions (first two layers) can be equivalent to characteristics and subcharacteristics levels (with a user's viewpoint), while the Metrics are referred in a more technical context.

⁴ The brackets for “Metric” on the third table row indicate that the model architecture does not formally mention that layer, even if it exists and is needed to make evaluations.

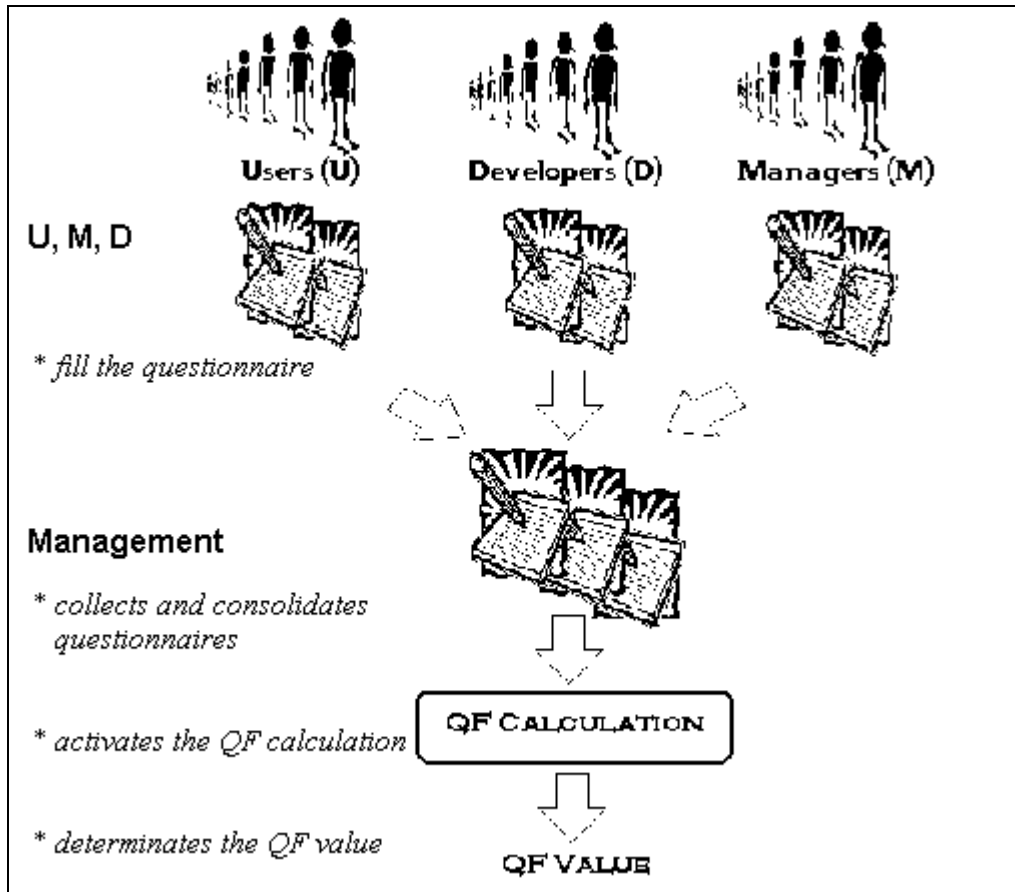


Figure 1 - QF procedure flow

4. Quality Factor: the calculation

Now it is time to present the design of the procedure to calculate QF, which returns a value which integrated the users', developers' and managers' opinions about the project being measured. The measurement instruments needed to calculate QF⁵ are:

- **questionnaire** (Form Q1): it is the technical instrument we use to collect data. Using the samples analysis through the stratified sample technique with Universe sensible elements divided in non overlapping groups (*strati*) and then choosing a random sample in every stratus we can have a good sample representativity and reliability of the values found. The questionnaire is structured following Statistics and Social Research Methodology principles, handed over to the interviewed person and filled up by him/her without any help from an interviewer (6).
- **a list of Tables**⁶ (Form Q2) - that you can simply automate with a spreadsheet:
 - ◊ *Table A*: it contains the checklist for selection of most relevant project quality subcharacteristics and reports the questionnaire results
 - ◊ *Table B*: it is the QF Calculation Table, used in conjunction with
 - ◊ *Table C*: it is used for the Characteristics Priority Weight Calculation
 - ◊ *Table D*: it lists the generic weights for each rank position

To obtain QF value, once the questionnaires have been filled, a 6-step procedure is executed:

- 1) **Most important quality subcharacteristics determination.** The series of most important attributes according to interviewed people is determined through the fourth section of the questionnaire. In question no. 11, the ISO rating levels (10, 13) listed in Table 2 are used.

⁵ At the web address: <http://space.tin.it/scienza/luigibug/qf/qf.htm> it is possible to find the tables, the questionnaire and the detailed rules for the QF calculation.

⁶ The forms respect and are in line with Basili & Weiss suggestions (3) about the iterative process to design and test them and the three validation principles: correctness, consistency and completeness.

MARK	RATING	GLOBAL RATING
3	Excellent	Satisfactory
2	Good	
1	Fair	
0	Poor / Absent	Unsatisfactory

Table 2 - ISO 9126 Rating Levels

Filling out Table A with your results, the mean value for each subcharacteristic is obtained as well as the acceptability per cent level (equal to the number of people who have chosen that subcharacteristic on the total of interviewed ones). A mean value is needed in order to decide which are the most relevant subcharacteristics in the joint opinion of the three groups. Table A column 11 values must be copied in Table B column 9 only for the selected subcharacteristics.

- 2) **Single characteristics priority determination.** After having determined the most relevant subcharacteristics and their weight, the characteristic priority (between 1 and 6) is determined according to the decreasing number of subcharacteristics chosen in the individual questionnaires. When two or more characteristics have the similar priority, one of the following options can be chosen :

- the total number of subcharacteristics is equal for the characteristics – assigning the same rank position. For example *reliability* and *usability* have both rank 3. Next characteristics in the order will have rank 5;
- the total number of subcharacteristics is different for the characteristics - consider the respective characteristic percentage. For example, *reliability* and *functionality* have both rank 3. You will give rank 3 to *reliability* and rank 4 to *functionality* because $(n/3 > n/5)$, where n indicates the number of subcharacteristics chosen on the total of each characteristics (3 and 5 in the order);
- the total number of subcharacteristics is different for some of the characteristics - in this case you will consider first criteria a) and then criteria b). For example if *reliability*, *maintainability* and *portability* have rank 3, the final rank will be: *reliability* rank 3, *maintainability* and *portability* rank 4. Next characteristic will have rank 6.

- 3) **Characteristics weights assignment.** After the prioritisation of chosen characteristics, a corresponding weight must be associated to each rank position, for each group of interest (Table D) to obtain at the end a QF value in the desired range (p.e. 0-100). In the case of same ranks, the weight is equal to the arithmetical mean of the weights, because they must be conventionally equally spaced:

$$p_1 - p_2 = p_2 - p_3 = p_3 - p_4 = p_4 - p_5 = p_5 - p_6 = p_6 - 0 = p_6 \quad (\text{Equation 1})$$

with

$$p_{\max} = p_1 \text{ and } p_{\min} = p_6 > 0 \quad (\text{Equation 2})$$

In this way the TCV (Total Characteristic Value) maximum value is not altered. It is possible to determine the generic weights of Table D column 2 as follows, starting from the rules on step 2. The hypothesis of TCVmax happens in the situation shown in the following table, when all the 21 subcharacteristics have been selected in the filling of questionnaires:

CHARACTERISTIC [1]	NO. OF SUBCHAR. [2]	PRIORITY (U,M,D) [3]	SCV [4]	PRIORITY WEIGHT [5]	CV [6=4*5]
Functionality	5	1	15	p_1	$15p_1$
Reliability	3	4	9	$(p_4+p_5)/2$	$(p_4+p_5)*9/2$
Usability	3	4	9	$(p_4+p_5)/2$	$(p_4+p_5)*9/2$
Efficiency	2	6	6	p_6	$6p_6$
Maintainability	4	2	12	$(p_2+p_3)/2$	$(p_2+p_3)*12/2$
Portability	4	2	12	$(p_2+p_3)/2$	$(p_2+p_3)*12/2$

Table 3 - Weights for the TCVmax Calculation

from which:

$$TCV = \sum_{i=1}^6 CV_i \quad (\text{Equation 3})$$

and so

$$TCV_{\max} = 15p_1 + \left[\frac{9}{2}(p_4 + p_5) \right] 2 + 6p_6 + \left[\frac{12}{2}(p_2 + p_3) 2 \right] \quad (\text{Equation 4})$$

Because

$$p_n = p_6(6 - n + 1) \quad \text{with} \quad 1 \leq n \leq 6 \quad (\text{Equation 5})$$

it is possible to express TCV through p6:

$$TCV_{\max} = 15(6p_6) + 9(3p_6 + 2p_6) + 6p_6 + 12(5p_6 + 4p_6) \quad (\text{Equation 6})$$

and so

$$p_6 = \frac{TCV_{\max}}{249} \quad (\text{Equation 7})$$

It will be sufficient to substitute in (Equation 7) the TCV value with absolute maximum weight you want to fix for the qualitative evaluation. For instance, for a TCVmax=100, we will obtain:

$$p_6 = \frac{100}{249} = 0.401606425 \quad (\text{Equation 8})$$

Consequently Table D should become:

RANK	WEIGHT
1	2.40963855
2	2.008032125
3	1.6064257
4	1.204819275
5	0.80321285
6	0.401606425

Table 4 - Priority Weights Table

The sum of Table C columns 3-5-7 values must be copied in Table C column 8. At last, in Table C, column 9 you insert the result of the division of column 8 by 3 (the number of the viewpoints examined); that value must be reproduced in the corresponding row of Table B column 11 ("Priority Weight" column).

- 4) **Sum of subcharacteristics values (SSV).** put in Table A column 10 the sum of subcharacteristics values (column 9) for each characteristic:

$$SSV = \sum_{i=1}^n SCV_i \quad (\text{Equation 11})$$

where n is a variable number of subcharacteristics between 2 (*Efficiency*) and 5 (*Functionality*).

- 5) **Characteristics value (CV) calculation.** multiply column 10 by column 11 values in Table B and put them in column 12 .
- 6) **Total Characteristics Value (TCV) calculation.** This value is given by the sum of the six weighed characteristics values in column 12 and represents the esteemed quality level acknowledged to the project.

$$TCV = \sum_{i=1}^6 CV_i \quad (\text{Equation 12})$$

Finally the Quality Factor is expressed by the following formula:

$$QF = \frac{TCV}{TCV_{\max}} \quad (\text{Equation 13})$$

4. Summary

The proposed model starts from the need to obtain an integrated software quality measurement, which includes the different aspects - technical, economic and social - that live together in every organization, which however is not often presented in a unitary view. Starting from a questionnaire

filled out individually, gathering the opinions of users, developers and managers about the quality perceived in a certain product, it is possible to obtain a single numerical value useful at the management level to take economical decisions for the future. The proposed checklist is based on the current ISO/IEC 9126 standard and the procedure is in line with upcoming revisions of this standard. We also stressed the importance of standards-based measures - in particular those by ISO - within quality models. The instruments used for the surveys and the observation were obtained through field tests at Italian companies in the air transport area. QF can be used both as a single quality measure and in conjunction with quantitative evaluations as in (2, 7, 8).

5. References

- (1) ABRAN A., *Quality – The Intersection of Product and Process*, The 6th IEEE International Software Engineering Standard Symposium (ISESS'95), Montréal, Québec, Canada, (August 21-25 1995)
- (2) ABRAN A. & BUGLIONE L., *Implementation of a Three-Dimensional Software Performance Measurement Model*, Technical Report, Université du Québec à Montréal (UQAM), to be published (1999)
- (3) BASILI V.R. & WEISS D.M., A Methodology for Collecting Valid Software Engineering Data, *IEEE Transaction on Software Engineering*, Vol. SE-10, No. 6, pp. 728-738 (November 1984)
- (4) BASILI V.R. & ROMBACH H.D., The TAME Project: Towards Improvement-Oriented Software Environment, *IEEE Transaction on Software Engineering*, , Vol. 14, No. 6, pp. 758-773 (June 1988)
- (5) BOEHM B.W., BROWN J.R., LIPOW H., MACLEOD G.J. & MERRIT M.J., *Characteristics of Software Quality*, Elsevier North-Holland (1978)
- (6) BUGLIONE L., *Graphical User Interface (GUI): vantaggi tecnici, economici e sociali*, Tesi di Laurea, Università “La Sapienza”, Roma, Italia (1995)
- (7) BUGLIONE L. & ABRAN A., *Multidimensional Software Performance Measurement Models: A Tetrahedron-based Design*, Proceedings of the 8th International Workshop on Software Measurement, University of Magdeburg / CIM, Magdeburg, Germany, (17-18 September 1998)
- (8) CAVALLO A. & BUGLIONE L., *A 3D Software Productivity Measurement Model*, Software Quality Engineering Conference (SQE 97), CMP, Southampton, pp. 191-200 (1997)
- (9) DROMEY R.G., A Model for Software Product Quality, *IEEE Transactions on Software Engineering*, Vol. 21, No. 2, pp. 146-162 (February 1995)
- (10) FUSANI M., Quality Models for Software Evolution Instruments, *International Seminar on Software Measuring & Testing*, IEI-CNR/Qualital/SSSUP “S.Anna”, Pisa, Italia, (12 December 1995)
- (11) IEEE, *Std 1061-1992: Standard for a Software Quality Metrics Methodology* (1992)
- (12) ISO, *International Standard 8402: Quality - Vocabulary*, (1986)
- (13) ISO/IEC, *International Standard 9126: Information Technology - Software product evaluation – Quality characteristics and guidelines for their use* (1991)
- (14) ISO/IEC JTC1/SC7/WG6 N1752, *CD 9126-1:Information Technology - Software Quality Characteristics and Subcharacteristics - Part 1: Quality Characteristics and Subcharacteristics*, 17/7/97
- (15) ISO/IEC JTC1/SC7/WG6 N1669, *PDTR 9126-2:Information Technology - Software Quality Characteristics and Subcharacteristics - Part 2: External Metrics*, 30/6/97
- (16) ISO/IEC JTC1/SC7/WG6 N1713, *WD 9126-3:Information Technology - Software Quality Characteristics and Subcharacteristics - Part 3: Internal Metrics*, 16/5/97
- (17) ISO/IEC JTC1/SC7/WG6, *DIS 14598-1: Information Technology - Software product evaluation – Part 1: General Overview*, 3/6/97
- (18) MCCALL J.A., RICHARDS P.K. & WALTERS G.F., *Factors in Software Quality, Voll. I, II, III: Final Tech. Report*, RADC-TR-77-369, Rome Air Development Center, Air Force System Command, Griffith Air Force Base, NY (1977)
- (19) NATALE D., *Qualità e quantità nei sistemi software: teoria ed esperienze*, Franco Angeli Informatica (1995)
- (20) PRESSMAN R., *Software Engineering: a beginner's guide*, McGraw Hill (1988)