Computer Standards & Interfaces 48 (2016) 10–18

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

An analysis of the factors determining software product quality: A comparative study

Karina Curcio, Andreia Malucelli, Sheila Reinehr *, Marco Antônio Paludo

Graduate Program in Computer Science (PPGIa), Pontifical Catholic University of Paraná — PUCPR, Curitiba, Brazil

ARTICLE INFO

Article history:
Received 12 June 2015
Received in revised form 28 November 2015
Accepted 3 April 2016
Available online 7 April 2016

Keywords:
Software quality
Survey
Factors determining quality
ISO/IEC 9126
ISO/IEC 25000

ABSTRACT

There has been a great deal of research on software quality, but few studies have stressed the factors beyond the scope of software products that can influence the final product's quality. These factors can also determine project success.
Objective: In this paper, a comparative study is conducted of the determinants of software quality, based on a prior study that only explored U.S. CIOs' (Chief Information Officers) perceptions of factors that could affect the final quality of software products. The aim of this study is to explore the perceptions of different users involved in the software development cycle and generate results that can be generalized and employed as an aid in the management of software project resources.
Method: The study was conducted through an online survey to various users involved in the software development cycle in Brazil. The respondents analyzed the same 24 items proposed in the previous study, categorized into individual, technological, and organizational factors. Based on the 175 responses obtained, a factor analysis technique was applied, considering the statistical model of the main components in order to identify the factors determining the quality of software products.
Results: After the factor analysis, it was identified that all 24 analyzed items displayed factor loadings greater than 0.5. Nine factors (9 eigenvalues greater than 1.0) were extracted from this analysis, with the value of the total variance equal to 72%.
Conclusion: Based upon the comparison between the studies, it was concluded that the most relevant factor identified in both surveys presented an individual character. This factor related items such as competence, training, knowledge, and level of user involvement as well as resistance to change. It was also identified through factor analysis that technological aspects had the highest ratings due to the strong relationship of the items comprising these factors compared to organizational aspects.

1. Introduction

According to the study conducted by Cap Gemini, Sogeti, and HP, disclosed in the "World Quality Report 2014–15" [1], investments in the area of quality assurance have increased in recent years. The percentage of budget invested in this area increased from 18% in 2012 to 23% in 2013, and reached 26% in 2014. However, even with this obvious growth of investment, only one group (1% to 3%) of executives surveyed in 2012 and 2013 reported that their companies used more than 40% of their IT budget for quality assurance. However, even with the increased investment in the area of quality assurance, there is still no guarantee of the quality of the developed products. It should be noted that much of the success of software projects relates to user satisfaction and, consequently, the quality of the generated products. To address these software quality issues, the ISO/IEC (International Standards Organization/International Electrotechnical Commission) published the 25,000 family of standards known as SQuaRE (Software Product Quality Requirements and Evaluation), which presents the Model of Software Product Quality [2]. This model is based on software product quality characteristics and sub-characteristics that can be used for both specifying software quality requirements as well as for their evaluation. Due to the importance of this topic, several studies have been developed in the area of quality that explore software product quality characteristics and sub-characteristics [3]. However, most of the studies address purely technological aspects such as: Metrics to assess the functional quality of the products generated [4], the quality of the generated code presented [5,6], and the number of errors found or aspects related to the product's usability [7,8]. Little research has focused primarily on the analysis of behavioral aspects that could affect the quality of software products, with Hoffman [9] and Acuña et al. [10] being among the few examples.

After identifying this gap, a study was conducted in 2010 by Gorla and Lin [11], with the main objective of identifying the factors beyond the scope of the software product that could influence software quality in organizations. These factors could be organizational, technological, or individual. To conduct this study, the authors sent a survey to some American CIOs (Chief Information Officers). The choice of these respondents was motivated by a prior study [12] that indicated improved

* Corresponding author.
E-mail addresses: karina.curcio@gmail.com (K. Curcio), malu@ppgia.pucpr.br (A. Malucelli), sheila.reinehr@pucpr.br (S. Reinehr), marco@paludo.com (M.A. Paludo).

http://dx.doi.org/10.1016/j.csi.2016.04.002
0920-5489/

quality in information technology as one of the top five concerns of IT executives. At the end of the study, 112 responses were obtained, which were assessed using different methods of analysis such as factor analysis and logistic regression. The result of the first analysis was intended to derive the factors (individual, technological, and organizational) that influence the quality of software products. The result of the second analysis sought to measure the strength of the association between the factors and attributes of software quality. The results of these analyses could help CIOs and CEOs (Chief Executive Officers) in the development of quality improvement programs, enabling a suitable management of resources within an organization.

In the study conducted by Gorla and Lin [11], it was possible to derive the factors determining software quality and to identify which of those identified factors were predominant. The authors identified the "Capacity of the users" as the most representative factor of the first analysis (factor analysis), i.e., a factor classified as individual. However, at the end of the second analysis (logistic regression), factors related to organizational aspects were identified as the most influential for software quality. This result is not surprising, given that the respondents were all CIOs who, in general, have a strategic view of the business, thereby considering organizational factors to be more relevant than technological ones.

Due to the importance of identifying the factors determining software quality in order to ensure a better management of resources within organizations, this study aims to conduct a comparison with the results obtained in the first stage of analysis performed by Gorla and Lin, while using different profiles of respondents involved in the software development cycle. It will thus be possible to complement the prior study, which only explored the perceptions of American CIOs, by adding the perspectives of new respondents.

The inclusion of new respondents when compiling the results is relevant because it will thus be possible to try to generalize the results and reduce the bias generated in the previous study. To this end, the same data collection method (survey) will be used and the same analysis (factor analysis) will be performed, as proposed by Gorla and Lin [11], but from the perspective of new respondents.

This paper is organized as follows: Section 2 presents the literature review and an explanation of the ISO/IEC 9126 and ISO/IEC 25000 standards. Section 3 describes the model used in the research as well as the relationship between the variables. Section 4 presents the research method used and the statistical calculations in detail. Section 5 presents the analysis of the results and Section 6 discusses the results achieved. Section 7 concludes by presenting some limitations and possibilities for future work.

2. Theoretical background

2.1. History of software quality models

According to the ISO/IEC 8402 standard [13], which is referenced in the ISO/IEC 9126 [3] and ISO/IEC 25000 [2] standards, software quality is the software product's ability to satisfy explicit and implicit needs under specific conditions. Since software quality is considered multidimensional, it is very important to establish which aspects are important to evaluate.

Various software quality models have previously been proposed such as the McCall model in 1977 [14], followed by the Boehm model in 1978 [15], the FURPS model proposed by Robert Grady in 1987, and the Dromey model in 1995 [16].

The model proposed by McCall [14] in 1977 is considered one of the forerunners, originally emerging as a product quality improvement project, developed by the US Air Force Electronic Systems Division (ESD), the Rome Air Development Center (RADC), and General Electric. Initially established with 55 characteristics, the model was reduced to only 11 factors: correctness, reliability, efficiency, integrity, usability, maintainability, testability, flexibility, portability, reusability, and interoperability. The model organizes the quality characteristics according to three different aspects: product operation, product review, and product transition. The major contribution of this model is the relationship between the quality factors and metric qualities of software.

In the Boehm model [15], despite a close similarity to the McCall model, a hierarchical division of quality characteristics was proposed in order to further refine the model. As in the McCall model, Boehm also included the needs of users and added some other characteristics.

The FURPS model [17] was proposed by Robert Grady and the Hewlett-Packard Company. The quality characteristics were divided into two groups: functional and non-functional; therefore, the first letter of the acronym FURPS represents the functional characteristics of the model. The rest of the acronym "URPS" represents non-functional characteristics (usability, reliability, performance, supportability). IBM Rational Software used this model and later transformed it into the FURPS + model [18].

The main objective of the model proposed by Dromey [16] was to be comprehensive enough to work with different systems. Dromey believed that quality evaluation differed for each product and, therefore, a dynamic process was necessary. The model focused on the relationship of characteristics and sub-characteristics of quality, proposing sub-levels of relationships. The main characteristics of quality proposed by this model were functionality, reliability, maintainability, reusability, and portability.

In 2001, the ISO standardized the concept of software product quality and published the ISO/IEC 9126 standard. This standard is divided into four parts:

a) ISO/IEC 9126-1 product quality model;
b) ISO/IEC 9126-2 external metrics;
c) ISO/IEC 9126-3 internal metrics;
d) ISO/IEC 9126-4 quality in use.

Through the standard, six characteristics were specified for the software product quality model: functionality, reliability, usability, efficiency, maintainability, and portability. According to the ISO/IEC 9126-1 standard, the quality of the process contributes to improving the quality of the product, and the product contributes to improving the quality in use, as shown in Fig. 1. The software product's quality can be assessed by measuring the internal attributes (typically, static measurements of intermediate products), external attributes (typically by measuring the behavior of the code when executed) and, finally, the attributes of quality in use [3].

Due to the importance of these standards and the wide adoption of their use, they are constantly being reviewed. Subsequently, a new series of standards was created by the ISO/IEC, called SQuaRE (Software Product Quality Requirements and Evaluation), which became known as the ISO/IEC 25000 family of standards. This standard was divided into five parts:

a) ISO/IEC 2500n — quality management;
b) ISO/IEC 2501n — quality model division;
c) ISO/IEC 2502n — quality measurement division;
d) ISO/IEC 2503n — quality requirements division;
e) ISO/IEC 2504n — quality assessment division.

The overall objective of creating a set of SQuaRE standards was to obtain a logically organized, rich, and unified series covering two main processes: the specification of software quality requirements and the evaluation of software quality, supported by a process measuring software quality [2].

This set of standards formed the conceptual basis used to guide the concepts of software product quality in the study conducted by Gorla and Lin [11] as well as this study.