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

A B S T R A C T

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.

© 2016 Elsevier B.V. All rights reserved.

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).

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.
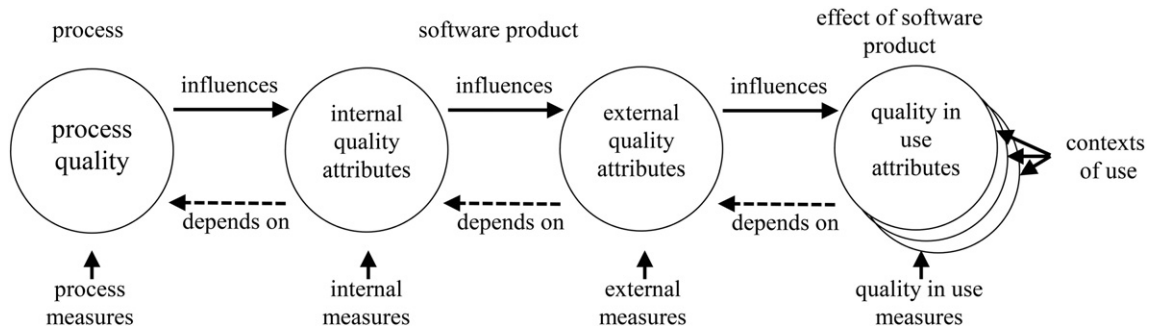
Fig. 1. Quality in the life cycle according to the ISO/IEC9126 standard [3].

2.2. Factor analysis

Factor analysis, one of the multivariate statistical methods, emerged in 1904 through a study by the American psychologist Charles Spearman concerning mental abilities [22,23]. As a result, he developed the technique of factor analysis, which tries to identify factors that explain the correlations among a set of variables. The purpose of applying factor analysis can be data reduction or structural simplification, classification and grouping of data, an investigation of the dependencies between variables, prediction, or the elaboration of hypotheses to then be tested. Factor analyses can achieve their goals through an exploratory or confirmatory perspective and to perform it is necessary to follow seven basic steps [19,20]. The first one collects measurements using a set of p variables, with n observations for each variable, in order to obtain the following arrangement of variables: $[X_{ij}]$ where ($I = 1,2, ..., n$) and ($j = 1,2, ..., p$).

The factor analysis model assumes that each variable X_j, extracted from a population with mean vector μ and covariance matrix Σ, is linearly dependent on a few random unobserved variables, $F_1, F_2, ... F_m$ ($m < p$), called common factors, and p additional sources of variation $\varepsilon_1, \varepsilon_2, ..., \varepsilon_p$, called errors or, sometimes, specific factors. The coefficient "l" is called the loading factor and represents the weight of the variable "i" in the factor "j" [23]. Therefore, the factor model can be represented by: $X_p = \mu_p + l_{p1}F_1 + l_{p2}F_2 + ... + l_{pm}F_m + \varepsilon_p$.

The second step obtains the correlation between the variables and for this correlation matrices or a covariance matrix can be used. It is also possible to extract other statistical data based upon the correlation and covariance matrices, as in the case of the eigenvalues. The purpose of calculating the eigenvalues is to obtain a set of independent vectors explaining the maximum data variability. The sum of all of the eigenvalues equals the total number of variables and indicates the total variance caused by each vector.

Usually, the acceptable and recommended values in exploratory research for the total variance are those greater than 80%, but it is not uncommon that in the social sciences, a field in which factor analysis is heavily used, values that explain 60% of the total variance are considered [21]. Other statistical data that can be extracted are loading factors, which determine the correlations between the original variable and the factor, the higher the loading factor, the greater the correlation with the factor. Usually, the relevant loading factors are those with values above 0.5 [20]. The third step selects the number of factors for inclusion in the analysis. There are various criteria to determine the correct number of factors to be extracted from an analysis, but in this study, the data analyses will be based solely on the criterion of eigenvalues greater than 1.0, as presented by Kaiser–Guttman [20,21]. The fourth step is the extraction of factors. There are different methods to perform the extraction of factors from the correlation matrix, but this study will address the method of principal components. The purpose of factor extraction is to find a set of factors that form a linear combination of the original variables or of the correlation matrix. Thus, if the variables $X_1, X_2, X_3, ...,$

X_n are highly correlated, they will be combined into one factor, and so on with the other variables of the correlation matrix. In other words, a linear combination of the variables $X_1, X_2, ..., X_n$ may be represented by the following formula: $F_j = C_{1j}X_1 + C_{2j}X_2 + ... + C_{nj}X_n$ where Fj is called the main component. Then, the variance explained by the first factor is excluded from the correlation matrix, with the result of residual matrices. Repeating the same steps above, the second main factor will be found, and so on, until a very small variance remains with no explanation. The fifth step is the rotation of the factors to assist the data interpretation task. There are two ways to accomplish the rotation: Orthogonal rotation (varimax rotation), which keeps the factors non-related, and oblique rotation, which correlates the factors among themselves. In this study, orthogonal (varimax) rotation will be used in order to maintain the same methods used by Gorla and Lin [11], enabling a comparative study. The sixth step is the interpretation of the factors. This task is a very subjective and depends much more on the conceptual background of the analyzer (versus an empirical one) [20]. The most important results extracted from a factor analysis are the factors; how many they are and which original variables are parts of each factor. In addition, the eigenvalues associated with each factor, and the percentage of total variance explained by each, demonstrate the degree of importance of each factor in explaining the proposed problem. The seventh and last step is the construction of factors scores in case to perform additional analysis. These scores can be obtained as a result of a linear combination of all the measures, weighted by the correspondent factor loading [19].

Once the steps and their respective details needed to promote the factor analysis are presented, in the next section, the basic structure of the research used by Gorla and Lin [11] will be presented, which will serve as a reference for this study.

3. Basic structure of the research

This section presents the basic structure of the research conducted by Gorla and Lin [11], also used in this study to make possible the comparative study. Fig. 2 illustrates how the relationship of the organizational, technological, and individual factors with software quality was conceived.

In the research conducted by Gorla and Lin [11], the software quality was represented by five characteristics, identified by: (1) reliability, (2) ease of use, (3) maintainability, (4) usefulness, and (5) relevance. According to the authors, the original characteristic of the ISO/IEC 9126 standard "functionality" was replaced by "relevance" and "usefulness," because a system rich in functionality will provide information that is more appropriate (relevant and useful) to users.

The characteristic of "usability," since it is linked to the ease with which the user implements a tool, was renamed by the authors "ease of use," based upon the assumption that in this way, the term would be clearer. The characteristics of reliability and maintainability were retained pursuant to the ISO/IEC 9126 standard.
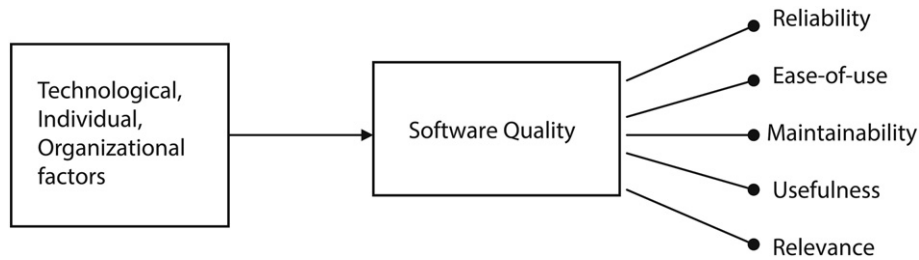
Fig. 2. Basic structure of the research proposed by Gorla and Lin [11].

This model did not include the characteristics of portability and efficiency from ISO/IEC 9126, since it was considered that the former should only be applied to software products that need to be implemented on multiple platforms, and that the latter should be considered a characteristic of quality internal to the product, and therefore related to response time and the consumption of computing resources.

The mapping between the characteristics of software product quality from ISO/IEC 9126 and those used by Gorla and Lin [11] is presented in Table 2.

Through this basic structure, Gorla and Lin [11] proposed a relationship between the factors (technological, organizational, and individual) and software quality to analyze the existence of a relationship between them, along with mapping the intensity of the strength of these relationships. To conduct this analysis, Gorla and Lin used independent and dependent variables. Conceptually, independent variables are the variables to be manipulated, introduced purposefully in order to verify their behavioral relationship to the other variables. The dependent variables are only measured or recorded, since their behavior is verified in accordance with the dependent variables.

Following this reasoning, the independent variables in the structure proposed by Gorla and Lin [11] are the organizational, technological, and individual factors. The dependent variables are the characteristics of software quality, represented here by reliability, ease of use, maintainability, relevance, and utility, with their behavior mapped according to the dependent variables.

The relationship between the independent variables (organizational, technological, and individual factors) and the dependent variables (characteristics of software quality) was proposed by Gorla and Lin [11] through the hypotheses described below:

Hypothesis 1. Technological factors, such as the use of a specific database, a programming language, and an appropriate developmental method, can directly affect the quality of a software product. It is understood that the inclusion of a process by means of a suitable development method may facilitate the maintainability of the software product and reduce the likelihood of errors. The use of a suitable development tool, presenting facilities for building graphical user interfaces, may result in products with interfaces which are rich in resources, easier to use, and much more user-friendly.

Hypothesis 2. Individual factors, such as user training, competence, resistance, and involvement, can affect the quality of a software product.

Table 2
Mapping of the characteristics of quality.

Attributes of quality from the ISO/IEC 9126 standard	Attributes used by Gorla and Lin [11]
1 — Reliability	1 — Reliability
2 — Usability	2 — Ease of use
3 — Maintainability	3 — Maintainability
4 — Functionality	4 — Relevance
	5 — Usefulness
5 — Efficiency	–
6 — Portability	–

It is understood that users who are more involved with the project can be more collaborative and contribute more to the project requirements, just as competent and well-trained users can have a better understanding of the inputs and outputs of the system and, consequently, can collaborate in the survey of processes. The most resistant users tend to position themselves negatively with regard to change and hence become a problem. These users tend to request more and more changes to the system, increasing the chances of the tool becoming unreliable.

Hypothesis 3. Organizational factors, such as top management support regarding appropriate projects, and budget availability, can lead to the development of better products. The higher the management position, the greater the understanding of organizational needs and, therefore, the greater the likelihood of more relevant and pertinent information being added to the products.

Since the goal of this study is to explore the perceptions of different users involved in the software development cycle to assess the factors determining software product quality and ensure better resource management within organizations, the above hypotheses exemplify how each of the factors (technological, organizational, and individual) can influence the final quality of the software products.

4. Research method

A survey was conducted with three distinct stages: collection of personal identification data, collection of the characteristics of the organization where the respondent works, and the evaluation of 24 items that might or might not influence software product quality in the opinion of each respondent.

The target audience was professionals involved in the software development cycle, but specifically those handling operational activities, with less involvement in management activities. This made it possible to compare the results with the research conducted by Gorla and Lin [11], which only involved CIOs who, in general, have a strategic view of the business.

The first step was to conduct a pilot test with two respondents in order to evaluate response time and the quality of the questions as well as to identify any weaknesses. After the pilot test, changes to some questions were necessary. The surveys were then sent to a small contacts network by e-mail and attached to it a message was sent asking them to spread the link to get more contributors for this research. The survey was available for 20 days and during this period this snowball technique was getting strength. A total of 175 responses were obtained from professionals of 62 different organizations in both public and private sectors, most of them from the southern of Brazil. These were then used in formulating this study's conclusions. During the execution of the survey some pre-defined options were offered for the respondents to be checked as a valid answer. One of these questions was about the role of the respondents in the software development cycle. In case the respondent did not find the answer there was an option called "Others" where the respondent should complete with an appropriate answer. Four out of 175 respondents complete this field with a different role as described here: Team Leader, Test Manager, Business
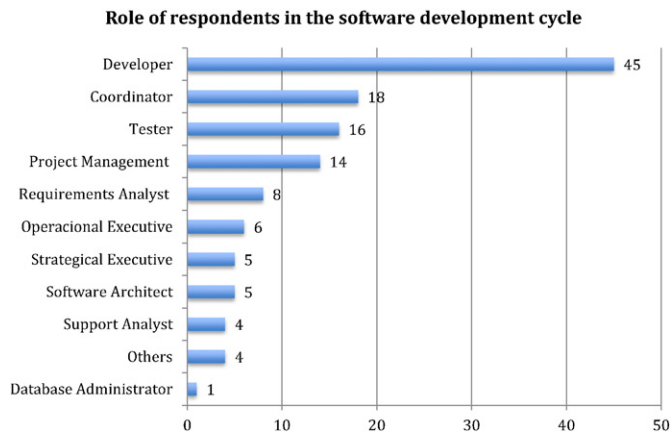
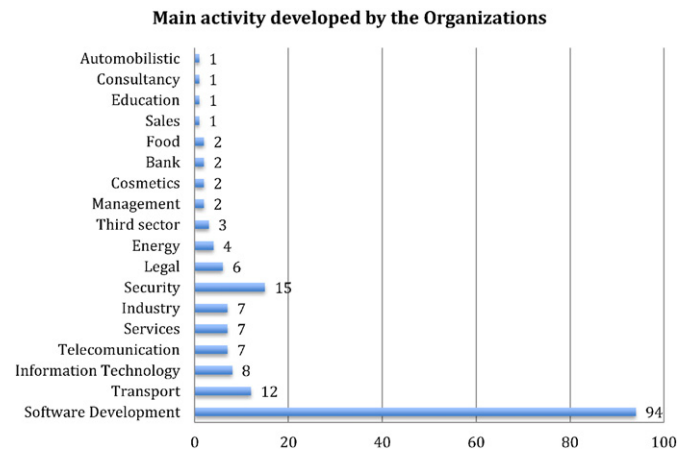**Fig. 3.** Graph representing the roles of the respondents.

Fig. 4. Graph representing the main activity developed by the organizations.

and Commercial. Fig. 3 shows the quantitative information describing the roles of the respondents involved in this study and Table 3 presents the profiles of the respondents.

Another important information provided by the respondents was the main activity of the organization where the respondents were working for at that moment. This information may be considered important as different types of companies have different characteristics, work with different scopes, have different kind of projects and consequently the employees get involved with it and focus their effort in different quality characteristics. Fig. 4 shows the quantitative results gathered during the application of the survey.

The 24 items to be evaluated were divided into categories: Individual, technological, and organizational (Table 4). To evaluate these, a Likert scale was used, asking the respondents to identify the degree of importance each entry had on the software product quality: "Very High," "High," "Medium," "Little," or "None."

In order to conduct a comparative study, taking into account the research conducted by Gorla and Lin [11], an exploratory factor analysis was performed regarding the set of answers to the 24 items in the questionnaire. The main components' model and the normalized varimax rotation were thus taken into consideration.

5. Results

Since the main objective of this study is to explore the perceptions of different users involved in the software development cycle in order to analyze, through a comparative study, the factors determining software product quality, an exploratory factor analysis was performed, taking into consideration the model of principal components and the normalized varimax rotation. The results from the questionnaires were described in frequencies and percentages, according to the response options (none, little, medium, high, or very high) as shown in Appendix A of this study.

The factors were extracted from the eigenvalues greater than 1, and for the composition of the factors, the items with factor loadings (weight) greater than 0.5, were included. The data were analyzed with IBM SPSS v.20 software. Nine factors were extracted (9 eigenvalues greater than 1), with an explained variance equal to 72%. The result of the factor analysis indicated that all the items defined by the model

Table 3
Respondent profile.

Characteristics	Frequency	Percentage
Type of the organization		
Public	76	43.42%
Private	99	56.57%
Length of time in this role (experience in years)		
Less than 1 year	14	8%
From 1 to 5 years	70	40%
More than 5 to 10 years	50	28.57%
More than 10 to 15 years	20	11.42%
More than 15 years	21	12%
Number of people in the organization		
Up to 9	10	5.71%
From 10 to 49	8	4.57%
From 50 to 99	7	4%
100 or more	150	85.71%
Number of individuals involved in the development of systems		
Up to 9	21	12%
From 10 to 49	15	8.57%
From 50 to 99	23	13.14%
100 or more	116	66.28%

Table 4
Items evaluated by the respondents.

Individual factors	Ind 1: user competency
	Ind 2: user training
	Ind 3: user knowledge
	Ind 4: user involvement
	Ind 5: user resistance to change
Technological factors	Tec 1: suitability of the Database Management System employed
	Tec 2: suitability of the programming language
	Tec 3: suitability of the development method employed in the organization
	Tec 4: support of the departments connected to IT
	Tec 5: experience of the team
	Tec 6: level of ability of the team
	Tec 7: type of system development methodology employed
	Tec 8: type of programming language employed
	Tec 9: type of model used by the Database Management System
Organizational factors	Org 1: support from management
	Org 2: managerial experience
	Org 3: sufficiency of budget
	Org 4: quality of the generated documentation
	Org 5: overall number of employees in the organization
	Org 6: number of employees working in the development sector
	Org 7: employee turnover in the organization
	Org 8: employee turnover in the development sector
	Org 9: frequency with which users request changes in the system
	Org 10: position of the Board of Directors in the corporate organizational chart

Table 5
Results of exploratory factor analysis.

Factor	Weight of the factors
Factor 1: capabilities of users (I)	Eigenvalue: 4.74; percentage of variance: 19.75%
Ind 1: user competency	0.776
Ind 2: user training	0.803
Ind 3: user knowledge	0.834
Ind 4: user involvement	0.731
Ind 5: user resistance to change	0.564
Factor 2: suitability of employed technology (T)	Eigenvalue: 2.53; percentage of variance: 10.55%
Tec 1: suitability of the Database Management System employed	0.871
Tec 2: suitability of the programming language	0.811
Tec 8: type of programming language employed	0.633
Tec 9: type of model used by the Database Management System	0.786
Factor 3: technological capacity (T)	Eigenvalue: 2.32; percentage of variance: 9.66%
Tec 5: experience of the team	0.823
Tec 6: level of ability of the team	0.861
Tec 7: type of system development methodology employed	0.582
Factor 4: size of the organization (O)	Eigenvalue: 1.75; percentage of variance: 7.30%
Org 5: overall number of employees in the organization	0.845
Org 6: number of employees working in the development sector	0.794
Factor 5: influence of upper management (O)	Eigenvalue: 1.50; percentage of variance: 6.25%
Org 1: support from management	0.770
Org 2: managerial experience	0.758
Org 10: position of the Board of Directors in the corporate organizational chart	0.561
Factor 6: maturity of the organization (T)	Eigenvalue: 1.26; percentage of variance: 5.26%
Tec 3: suitability of the development method employed in the organization	0.781
Tec 4: support of the departments connected to IT	0.599
Factor 7: stability of organization (O)	Eigenvalue: 1.12; percentage of variance: 4.66%
Org 7: employee turnover in the organization	0.881
Org 8: employee turnover in the development sector	0.751
Factor 8: quality of the service provided (O)	Eigenvalue: 1.05; percentage of variance: 4.37%
Org 4: quality of the generated documentation	0.573
Org 9: frequency with which users request changes in the system	0.798
Factor 9: available budget (O)	Eigenvalue: 1.01; percentage of variance: 4.19%
Org 3: sufficient budget	0.774

Table 6
Result of the internal consistency of the factors found.

Factor	Number of items	Cronbach's alpha
Factor 1 — capabilities of users	5	0.825
Factor 2 — suitability of the employed technology	4	0.829
Factor 3 — technological capacity	2	0.861
Factor 4 — size of the organization	2	0.707
Factor 5 — influence of the upper management	3	0.528
Factor 6 — maturity of the organization	3	0.579
Factor 7 — stability of organization	2	0.665
Factor 8 — quality of the service provided	2	0.422
Factor 9 — available budget	1	–

6. Discussion of the results

According to the data presented in Section 5, this study identified nine factors that determine software product quality: (1) capabilities of users, (2) suitability of the employed technology, (3) technological capacity, (4) size of the organization, (5) influence of the upper management, (6) maturity of the organization, (7) stability of organization, (8) quality of the service provided, and (9) available budget. For each factor, its percentage of variance was calculated, which represents the degree to which each factor is determinant and representative of software quality. The sum of the variances, calculated for each factor, provides a total percentage, which illustrates the degree to which the factors analyzed together are representative. Thus, in this study, the sum of the partial variances of the analyzed factors explains a variance of 72%.

In the study conducted by Gorla and Lin [11], which focused on CIOs, six factors determining software product quality were obtained: (1) capabilities of users, (2) attitude of the management, (3) stability of organization, (4) suitability of technology, (5) capability of IS department, and (6) responsiveness of IS department. These factors accounted for a variance of 64.43%. In order to conduct a comparative analysis, Table 7 shows the results obtained from the study conducted by Gorla and Lin [11].

Table 7
Results of the exploratory factor analysis of the study conducted by Gorla and Lin [11].

Factor	Weight of the factors
Factor 1: capabilities of users (I)	Eigenvalue: 4.40; percentage of variance: 25.85%
User competency	0.783
User training	0.713
User knowledge	0.595
User involvement	0.559
User resistance to change	0.540
Factor 2: attitude of management (O)	Eigenvalue: 1.70; percentage of variance: 10.00%
Sufficiency of budget	0.820
Quality of documentation	0.672
Support from management	0.503
Factor 3: stability of organization (O)	Eigenvalue: 1.47; percentage of variance: 8.62%
Turnover in IS department	0.847
Turnover in the company	0.839
Factor 4: suitability of technology (T)	Eigenvalue: 1.24; percentage of variance: 7.27%
Suitability of database management	0.823
Suitability of programming language	0.637
Factor 5: capability of IS department (T)	Eigenvalue: 1.12; percentage of variance: 6.61%
Suitability of development method	0.751
Support from IS department	0.602
Experience of IS staff	0.553
Factor 6: responsiveness of IS department (O)	Eigenvalue: 1.03; percentage of variance: 6.08%
Rank of IS director	0.774
Frequency of user's change request	0.678

showed a weight greater than 0.5, ranging from 0.561 to 0.881. Thus, no items were removed.

Table 5 presents the results of the items that comprise each of the factors evaluated in the survey. For each factor, the items that comprise it are listed, with their own weight and eigenvalue, with the variance explained by the factor.

To assess the internal consistency of each factor, Cronbach's Alpha coefficients were estimated [20]. This coefficient ranges from 0 to 1, and the higher the value, the greater the internal consistency. The Cronbach Alpha coefficient calculated for each factor describes the degree to which the items that constitute the factor measure the same concept (domain, construct). This can be interpreted as the degree to which the items of a factor are associated with each other. Table 6 shows the factors extracted from the analysis, the number of items that constitute each factor (with factor loadings greater than 0.5) and the values of Cronbach's Alpha coefficients found for each factor.

Interestingly, the most relevant factor identified in both studies was Factor 1 — capabilities of users. This factor represented a variance of 25.85% in the research conducted by Gorla and Lin [11] and 19.75% in the present study. In both surveys, Factor 1 was composed of the same items: (1) user competency, (2) user training, (3) user knowledge, (4) user involvement, and (5) user resistance to change, which further reinforce user influence on the perceived quality of software products.

Some recent researches [26–28] already showed that user involvement can influence projects and are considered as the key point for developing useful and usable systems. Early user involvement is related to the prevention of unneeded and expensive features, an increase of user satisfaction, better quality requirements, and consequently to successful projects. Companies should encourage all practitioners to increase user participation and involvement in all phases of their software development. As a result of these researches, companies should invest in training of their staff in order to provide favorable conditions to encourage the involvement and participation of users in all phases of software development.

In the study conducted by Gorla and Lin [11], the Cronbach's Alpha found for Factor 1 was 0.7217, while in the present study it was 0.825. The value of Cronbach's Alpha recommended by some researchers for exploratory research is at least 0.60 [20,21]. This shows that in both cases, values above the recommended value were achieved, highlighting the strength of the association between the items comprising this factor.

Factors 2 and 3 of this study (respectively, "suitability of the employed technology" and "technological capacity," classified as technological factors) differ from the study by Gorla and Lin in factors 2 and 3 ("attitude of management" and "stability of organization"), classified as organizational factors, ranked as more significant factors.

Factor 2, "suitability of the employed technology", presented a variance of 10.55% in this study, while the "attitude of management" presented a variance of 10.00% in the study by Gorla and Lin [11], i.e., the variance values were very close to each other.

Factor 3, "technological capacity," presented a variance of 9.66% in this study, while the "stability of organization" had a variance of 8.62% in the study by Gorla and Lin [11].

The results of the analysis of the factors 2 and 3 reinforce the idea that respondents with managerial profiles tend to give greater importance to organizational factors, while respondents with more operational profiles tend towards a greater appreciation of technological factors.

All items that composed Factors 2 and Factor 3 are recurrent concerns like suitability of the database management systems, programming language and system development methodology employed. Nowadays with the emergence of the startups concept in the market [29,30], such topics as agile methodologies for software development and agile development frameworks have become not only relevant as needed. Existing researches describe several benefits of using agile approaches as a powerful methodology that can enable teams to improve productivity, enhance visibility and achieve higher customer satisfaction [31,32]. These methods came to tackle requirements change quickly, satisfy customers, support interaction, communication and produce high quality products [33]. These new approaches are giving clear signs of real benefits and companies in general have to pay attention and invest on it.

Factor 4 was represented in this study as the "size of organization" and classified as an organizational item. This item showed a variance of 7.30%, while in the study by Gorla and Lin [11], Factor 4 "suitability of technology," classified as a technological factor, displayed a variance of 7.27%. At this point, despite having very close variance values, the perceptions regarding origins (technological, organizational) were again completely different. This strong difference of point of view may have appeared in consequence of the new market entrants already mentioned in the previous item.

The factors "influence of upper management" (Factor 5), "maturity of the organization" (Factor 6) and "quality of the service provided"

(Factor 8) of this study displayed Cronbach's Alpha values equal to 0.528, 0.579, and 0.422, respectively, values considered below the acceptable value of 0.60. This implies that the relationship between the items that comprise this factor is weak.

Factor 9 ("sufficiency of budget") of this study did not correlate in any way with the other items on the questionnaire and it can therefore be concluded that this item alone is a factor determining software quality.

This is probably a direct consequence of the economic situation of the country. Employees are having to deal with economic restrictions, cutting budgets in general, diminishing resources and increased risks in the projects. Since no relation to the other items of the questionnaire was identified, it was not possible to calculate Cronbach's Alpha for this factor. However, as its obtained eigenvalue was 1.01, and according to the Kaiser–Guttman criterion that validates eigenvalues greater than 1.0, this factor becomes representative for the study. The variance identified for this factor was 4.19%.

Another important fact that can be observed is that in the study by Gorla and Lin [11], of the 24 items analyzed in the questionnaire, only 17 were retained after the factor analysis. The items (1) type of programming language employed, (2) type of model used by the database management system, (3) type of development methodology employed, (4) overall number of employees in the organization, (5) number of employees working in the development sector, and (6) managerial experience were removed, because the results of the eigenvalues did not reach values greater than 1.0.

This might have occurred precisely because of the CIOs' lack of knowledge of operating activities in the study by Gorla and Lin [11], which finally led to results below the acceptable value.

In this study, no questionnaire item was discarded after the factor analysis, since all of the factors achieved representative values (greater than 1).

7. Conclusion

The main objective of this study was to explore the perceptions of different users involved in the software development cycle in the assessment of factors determining software product quality. This made it possible to conduct a comparative analysis of this study and the prior study by Gorla and Lin [11], which explored the perception of only one type of respondent. The results of the study by Gorla and Lin were based on 112 responses from American CIOs, related to information technology projects. The research presented here was based on 175 responses, of which 83.33% of the respondents were involved in operating activities during the software development cycle.

In the study by Gorla and Lin [11], with the target audience of American CIOs, the 24 analyzed items were summarized and grouped into six factors. There were three organizational factors (attitude of management, stability of organization, responsiveness of IS department), two technological factors (suitability of technology, capability of IS department) and one individual factor (capabilities of users).

In this study, after performing the calculations for factor analysis, the 24 items could be summarized and grouped into nine factors. There were five organizational factors (size of the organization, influence of upper management, stability of the organization, quality of the service provided, and available budget), three technological factors (suitability of the employed technology, technological capacity, maturity of the organization), and one individual factor (user competency).

Since the order of the factors represents their importance, according to which the factors are grouped, it is evident that in both studies, the individual factors were ranked as number one, i.e., the most prominent. It is therefore possible to conclude that, contrary to what was expected, the management respondents did not have organizational items as their main focus, just as operational respondents did not have technological items as their main focus.

Therefore, this study complements the results obtained in the previous one (which only explored one respondent profile) and thereby add new insights.

As a result of this study, it is possible to highlight the importance of the participation and influence of the end user in software product quality. In both surveys, the most representative factor in the factor analysis was identified as "User competency." This factor, in both surveys, was composed of the following items: (1) user competency, (2) user training, (3) user knowledge, (4) user involvement, and (5) user resistance to change.

As similarly revealed in the study by Gorla and Lin [11], the results obtained in this study through factor analysis are important to CIOs and CEOs, respectively, since they are responsible for managing the allocation of resources in organizations as well as for improving product quality.

The inclusion of different respondents allows the test results to be generalized, and different strategies to be adopted by corporate managers. According to the results from both studies, strategies could be adopted such as resource allocation for the training of end users, the purchase of equipment (hardware and software), or the training of a company's own employees.

7.1. Threats to the validity of the research

An important point to be analyzed, and which can be considered a threat to the validity of this research, is related to the final results obtained in the total variance. As shown in Section 2.2, the optimal value for the total variance of exploratory research is at least 80%, but many studies in the area of social sciences have been recognized and accepted with total variance values of 60%.

This research point of vulnerability was also previously stated in the study by Gorla and Lin [11], given that the six factors they found as determinants of quality software products accounted for only 64.43% of the total variance of the performed experiment. Other studies have demonstrated variance values between 80% and 88.95%. One example is the work of Lake and Cook [24], which identified five (or less) factors measuring the domain of complexity of programs oriented towards objects in a group of metrics to explain a total variance of 84.61% to 88.95%. Another example is the work of Hanebutte, Taylor, and Dumke presented [25], in which five factors related to the main internal software metrics explained a variance of 80%.

In this study, nine factors determining quality were found, representing a total variance of 72%. In this respect, this result was slightly better than the result of the study by Gorla and Lin [11], although there are indications that the very identification and classification of the analyzed items may present study weaknesses or be incomplete.

7.2. Limitations

The results of this study should be analyzed considering the limitations of the respondents' profiles. Even though respondents with different roles in the software development cycle were involved, it is important to emphasize that some of them also had managerial characteristics, as in the case of the coordinators, executives in the operational area (managers), and executives in the strategic area (presidents, vice presidents, and directors). However, this was not a very representative portion, accounting for only 16.67% of all respondents. The other 83.33% were respondents with operational profiles, who therefore complement the vision of the study by Gorla and Lin [11], where only the managerial profiles of American CIOs were explored.

Another important point to be evaluated is the understanding of the respondents in relation to the items evaluated. During the pilot phase of the survey some of the issues to be evaluated by the respondents were not well written and were causing doubts. Because of this a redesign of some questions and factors was necessary to be made for better understanding. Nevertheless it is not possible to guarantee that all respondents completely understood all factors or had the same understanding of the issues. As presented in this paper, the companies where the respondents develop their roles are quite different. They vary in type, in numbers of clients, in scope, types of projects and these points should be considered since their focus on the characteristic of quality are completely different.

7.3. Future work

As thoughts for future work, in order to complement the analysis by Gorla and Lin [11], further comparisons should be made to detect the extent to which the factors determining quality, already identified, relate to the attributes of software quality through logistic regression analysis. It would thus be possible to examine whether organizational factors actually have more influence on the quality of software products compared to individual or technological factors. Another proposal for future work would be to explore the different profiles of respondents, while targeting the questions; i.e., the questions related to organizational factors could be answered by management respondents, the questions related to technological factors could be answered by operational respondents, and the questions related to individual factors could be answered by both.

Appendix A. Frequency and percentage of agreement with each item and the response options

Item		Response options				
		None	Little	Medium	High	Very high
		n (%)	n (%)	n (%)	n (%)	n (%)
Factor 1						
I1	User competency	5 (2.9)	27 (15.4)	58 (33.1)	68 (38.9)	17 (9.7)
I2	User training	2 (1.1)	16 (9.1)	60 (34.3)	67 (38.3)	30 (17.1)
I3	User knowledge	3 (1.7)	14 (8.0)	51 (29.1)	78 (44.6)	29 (16.6)
I4	User involvement	1 (0.6)	6 (3.4)	30 (17.1)	73 (41.7)	65 (37.1)
I5	User resistance to change	1 (0.6)	11 (6.3)	45 (25.7)	64 (36.6)	54 (30.9)
Factor 2						
T1	Suitability of the Database Management System employed	2 (1.1)	17 (9.7)	49 (28)	86 (49.1)	21 (12)
T2	Suitability of the programming language	2 (1.1)	12 (6.9)	54 (30.9)	83 (47.4)	24 (13.7)
T8	Type of programming language employed	4 (2.3)	38 (21.7)	74 (42.3)	49 (28.0)	10 (5.7)
T9	Type of model used by the Database Management System	3 (1.7)	28 (16.0)	83 (47.4)	47 (26.9)	14 (8.0)

(continued on next page)

Appendix A (*continued*)

Item		Response options				
		None	Little	Medium	High	Very high
		n (%)	n (%)	n (%)	n (%)	n (%)
Factor 3						
T5	Experience of the team	0 (0)	1 (0.6)	18 (10.3)	89 (50.9)	67 (38.3)
T6	Level of ability of the team	0 (0)	0 (0)	10 (5.7)	101 (57.7)	64 (36.6)
Factor 4						
O5	Overall number of employees in the organization	23 (13.1)	56 (32.0)	59 (33.7)	30 (17.1)	7 (4.0)
O6	Number of employees working in the development sector	12 (6.9)	26 (14.9)	83 (47.4)	40 (22.9)	14 (8)
Factor 5						
O1	Support from management	0 (0)	2 (1.1)	29 (16.6)	83 (47.4)	61 (34.9)
O2	Managerial experience	1 (0.6)	7 (4.0)	39 (22.3)	77 (44.0)	51 (29.1)
O10	Position of the Board of Directors in the corporate organizational chart	6 (3.4)	24 (13.7)	66 (37.7)	51 (29.1)	28 (16.0)
Factor 6						
T3	Suitability of the development method employed in the organization	1 (0.6)	3 (1.7)	43 (24.6)	86 (49.1)	42 (24.0)
T4	Support of the departments connected to IT	0 (0)	9 (5.1)	48 (27.4)	87 (49.7)	31 (17.7)
T7	Type of system development methodology employed	2 (1.1)	11 (6.3)	56 (32.0)	77 (44.0)	29 (16.6)
Factor 7						
O7	Employee turnover in the organization	8 (4.6)	33 (18.9)	58 (33.1)	54 (30.9)	22 (12.6)
O8	Employee turnover in the development sector	0 (0)	12 (6.9)	30 (17.1)	77 (44.0)	56 (32.0)
Factor 8						
O4	Quality of the generated documentation	0 (0)	18 (10.3)	38 (21.7)	58 (33.1)	61 (34.9)
O9	Frequency with which users request changes in the system	3 (1.7)	8 (4.6)	47 (26.9)	65 (37.1)	52 (29.7)
Factor 9						
O3	Sufficiency of budget	0 (0)	0 (0)	30 (17.1)	71 (40.6)	74 (42.3)

References

[1] http://www.capgemini.com/resources/world-quality-report-2014-15.
[2] ISO/IEC 25010, International organization for standardization/international electrotechnical, Software Engineering — Software Product Quality Requirements and Evaluation (SQuaRE) — Quality Model, ISO/IEC, 2011.
[3] ISO/IEC 9126, Software Engineering – Product Quality – Part 1: Quality Model, International Standards Organization, Geneva, Switzerland, 2001.
[4] A. Carrillo, P. Mateo, M. Monje, Metrics to evaluate functional quality: a systematic review, Inf. Syst. Technol. (CISTI), 2012 7th Iber. Conf., 2012.
[5] S. Lauesen, H. Younessi, Is software quality visible in the code? IEEE Softw. (August 1998) 69–73.
[6] P. Correia, K. Schill, J. Visser, R.B. Jose, Standardized Code Quality Benchmarking for Improving Software Maintainability, 2012 287–307.
[7] A. Seffah, M. Donyaee, R.B. Kline, H.K. Padda, Usability Measurement and Metrics: A Consolidated Model, 2006 159–178.
[8] A. Abran, A. Khelifi, W. Suryn, Usability Meanings and Interpretations in ISO Standards, 2003 325–338.
[9] R. Hofman, Behavioral economics in software quality engineering, Empir. Softw. Eng. 16 (2) (2010) 278–293.
[10] S.T. Acuña, M. Gómez, N. Juristo, How do personality, team processes and task characteristics relate to job satisfaction and software quality? Inf. Softw. Technol. 51 (3) (2009) 627–639.
[11] N. Gorla, S.C. Lin, Determinants of software quality: a survey of information systems project managers, Inf. Softw. Technol. 52 (6) (2010) 602–610.
[12] J. Luftman, R. Kempaiah, Key issues for IT executives 2007, MIS Q. Exec. 7 (2008) 99–112.
[13] AS/NZS ISO 8402, Quality Management and Quality Assurance — Vocabulary, 1994.
[14] J.A. McCall, P.K. Richards, G.F. Walters, Factors in Software Quality, Griffiths Air Force Base, N. Y. Rome Air Development Center Air Force Systems Command, 1977.
[15] B.W. Boehm, J.R. Brown, M. Lipow, Quantitative evaluation of software quality, International Conference on Software Engineering, Proceedings of the 2nd International Conference on Software Engineering, 1976.
[16] R.G. Dromey, A model of software product quality, IEEE Trans. Softw. Eng. (February 1995) 146–162.
[17] R. Grady, D. Caswell, Software Metrics: Establishing a Company-wide Program, Prentice Hall, 1987 159.
[18] R. Grady, Practical Software Metrics for Project Management and Process Improvement, Prentice Hall, Englewood Cliffs, 1992.
[19] J. DeCoster, Overview of factor analysis, 1998 Retrieved 08, August, 2014 from http://www.stat-help.com/notes.html.
[20] J.F.J. Hair, Multivariate Data Analysis, seventh ed Upper Saddle River, Prentice Hall, 2010.
[21] R.F. DeVellis, Scale Development: Theory and Applications, Sage, Newbury Park, CA, 1991.
[22] F.H.C. Marriot, The Interpretation of Multiple Observations, Academic Press, New York, 1974 117.
[23] R.A. Johnson, D.W. Wichern, Applied Multivariate Statistical Analysis, sixth ed. Pearson Education, Upper Saddle River, 2007.
[24] A. Lake, C. Cook, Use of factor analysis to develop OOP software complexity metrics, Proc. 6th Annual Oregon Workshop on Software Metrics, Silver Falls, Oregon, 1994.

[25] N. Hanebutte, C.S. Taylor, R.R. Dumke, Techniques of successful application of factor analysis in software measurement, Empir. Softw. Eng. 8 (1) (2003) 43–57.
[26] S. Kujala, M. Kauppinen, L. Lehtol, T. Kojo, The role of user involvement in requirements quality and project success, Proceedings of the 13th IEEE International Conference on Requirements Engineering August 29–September 02, 2005, pp. 75–84.
[27] D. Pagano, B. Bruegge, User involvement in software evolution practice: a case study, Proceedings of the 2013 International Conference on Software Engineering, IEEE 2013, pp. 953–962.
[28] U. Abelein, H. Sharp, B. Paech, Does involving users in software development really influence system success? IEEE Softw. 30 (6) (2013) 17–23.
[29] E. Ries, The Lean Startup: How Constant Innovation Creates Radically Successful Businesses, Crown Business, 2011.
[30] S. Blank, Why the lean start-up changes everything, Harv. Bus. Rev. 91 (5) (2013) 63–72.
[31] K. Kaur, A. Jajoo, Manisha, Applying agile methodologies in industry projects: benefits and challenges, International Conference on Computing Communication Control and Automation 2015, pp. 832–836.
[32] A.M.M. Hamed, H. Abushama, Popular agile approaches in software development: review and analysis, International Conference on Computing, Electrical and Electronics Engineering 2015, pp. 160–166.
[33] H.P. Rakshit, A.P. Patil, An improvised perspective to agile methodology, IEEE 3rd International Conference on System Engineering and Technology, Aug. 19–20 2013.

Karina Curcio has a Bachelor's degree in Informatics, Master's in Electrical Engineering at Federal University of Technology – Paraná. She is a member of the Software Engineering Research Group at Pontifical Catholic University of Paraná and her research interests are in software quality, software process improvement and innovation.

Andreia Malucelli has a Bachelor's degree in Informatics, Master's in Electrical Engineering and Doctorate in Electrical and Computing Engineering. Professor at Pontifical Catholic University of Paraná, a Researcher at the Software Engineering Research Group, with experience in computer science, software engineering, artificial intelligence, organizational learning, ontologies, multiagent systems and healthcare information systems.

Sheila Reinehr has a Bachelor's degree in Mechanical Engineering, Master's in Informatics and Doctorate in Engineering. Professor at Pontifical Catholic University of Paraná, a Researcher at the Software Engineering Research Group, with experience in computer science, software engineering, software process improvement, software quality, project and portfolio management, software product lines, and metrics.

Marco Antônio Paludo has a Bachelor's degree in Computer Science, Master's in Applied Informatics and a PhD candidate at Pontifical Catholic University of Paraná, where he also is a lecturer and Project Manager with experience in computer science, software engineering, software process improvement, software quality and modeling.