Information and Software Technology 52 (2010) 602–610

Contents lists available at ScienceDirect

Information and Software Technology

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

Determinants of software quality: A survey of information systems project managers

Narasimhaiah Gorla [a,*], Shang-Che Lin [b]

[a] Department of MIS, American University of Sharjah, PO Box 26666, Sharjah, United Arab Emirates
[b] Department of Computer Information Systems, Cleveland State University, Cleveland, OH, USA

ARTICLE INFO

Article history:
Received 20 March 2009
Received in revised form 19 November 2009
Accepted 30 November 2009
Available online 4 December 2009

Keywords:
Factor analysis
Information systems projects
Software quality
Reliability
Maintainability
Usefulness

ABSTRACT

Software quality is important for the success of any information systems (IS). In this research, we find the determinants of software quality. We used five attributes for software quality: system reliability, maintainability, ease of use, usefulness, and relevance. By surveying 112 IS project managers, we collected data about their perceptions on the software quality attributes and their determinants. We arrived at six factors through exploratory factor analysis. We determined the individual factors that impacted the software quality attributes; for example, reliability is associated with responsiveness of IS department; ease of use is influenced by the capabilities of users and attitude of management; and usefulness is impacted by capabilities of IS department and responsiveness of IS department. We show that organizational factors are more important than technical factors in impacting software quality in IS projects. We provide implications of our research to practice and to future research.

1. Introduction

Billions of dollars are spent in software projects because their success is important to organizations, system departments, and system managers. An important aspect of system success is software quality – "Software quality can determine the success or failure of a software product in today's competitive market" [58]. Though 50 years of software development has come and gone, software quality today is no better than decades ago and is worse in some cases [47]. The cost of faulty software in the US runs into tens of billions of dollars and it represents approximately 1% of Gross Domestic Product [25]. It should be noted that a software product is not only a source code, but it includes what user sees [36]; software quality is more than static assessment and includes non-functional or behavioral attributes, such as reliability and maintainability [60]. Most software engineering research emphasizes technical rather behavioral aspects of software projects [59].

The increasing dependence of organizations on information systems and the increasing losses to organizations due to poor software drive management attention towards software quality improvement. A recent survey of Chief Information Officers (CIOs) indicates that "Improve IT quality" is one of the top concerns facing IT executives [38]. By properly leveraging software, organizations can achieve operational efficiencies by managing internal resources well. As software quality is a multidimensional measure, it is important to determine what aspects of software quality are critical to organizations. Though software quality problems have been acknowledged for several decades, the fact that software quality is of great concern in organizations even today implies that there is a need for more research emphasis to investigate the causes behind the poor software quality. In order to understand these critical issues, the important question we address in this research is *"What external factors influence software quality?"* These factors can be organizational, technological, or end user related. The answers to the above question will help CIOs to devise software quality improvement programs with which scarce resources are allocated more effectively. Furthermore, such programs should enhance software quality, thereby resulting in successful information systems and organizations.

Though software quality is an integral part of information system success [19], there have been few comprehensive studies on factors that influence software quality. Most studies addressed technical factors of software quality without adequately considering the user and organizational influence on software quality. Furthermore, quantitative survey-based research regarding software development is lacking [59]. In this research we start with variables related to individual or user characteristics, organizational or management characteristics, and technological characteristics. We derive the significant dimensions of these variables using exploratory factor analysis based on data collected from 112 information system (IS) projects through a sur-

* Corresponding author. Tel.: +971 6 5152306; fax: +971 6 5585065.
E-mail address: ngorla@aus.edu (N. Gorla).

0950-5849/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.infsof.2009.11.012

vey. We also assess the influence of the dimensions on software quality attributes. In our research, we use five software quality attributes. Our measures of software quality represent error proneness of system ('reliability'), effort required to use the system ('ease of use'), effort required to maintain the system ('maintainability), benefit of information output ('usefulness'), and suitability of information output to decision making ('relevance'). These are representative of commonly used attributes of software quality. The objectives of this research are:

1

(i) *to derive individual, organizational, and technological factors that influence software quality, and*

2

(ii) *to assess the strength of association between the above determinants and software quality attributes.*

The organization of this paper is as follows. Section 2 has theoretical perspectives on dependent variables (software quality attributes) and independent variables (individual, organizational, and technological factors). Section 3 contains the research model depicting relationship between the independent variables and the dependent variables. Section 4 provides research methodology where in the survey method is described, respondent and IS characteristics are outlined, and a statistical is presented. Section 5 has results in two parts: the factors of independent variables and the strength of relationships between these factors and software quality attributes. Section 6 has discussion of results followed by Section 7 that contains conclusion with implications to research and practice.

2. Theoretical perspectives

2.1. Software quality

As per ISO 9126 standard [29], quality is defined as "the totality of features and characteristics of a product or service that bears on its ability to satisfy given needs" [1]. Software quality has also been defined in terms of two types of product characteristics [10,11]: (i) external quality (how the product works in its environment), such as, usability and reliability, and (ii) internal quality (how the product was developed), such as, software structure and complexity [27]. In this research, we focus on external software quality attributes. Traditionally, software quality has been defined to be composed of correctness, reliability, usability, and maintainability [21]. Boehm [12] defines software quality to include portability, reliability, efficiency, human engineering, understandability, modifiability, and testability. The desirable software quality attributes for web applications are reliability, usability, security, availability, scalability, maintainability, and time to market [45].

Software quality plays a key role in information system (IS) success. DeLone and McLean proposed an IS success model [20] that consists of six interdependent measures of IS success: system quality, information quality, use, user satisfaction, individual impact and organizational impact. System quality and information quality are two major components of software quality. System quality is concerned with presence or absence of bugs in the system, consistency of user interface, ease of use, quality of documentation, and sometimes, quality and maintainability of program code [53]. Previous studies used the following measures for system quality: reliability and on-line-response time [57], reliability and ease of use [7], relevance, perceived usefulness [24], and maintainability [26]. Information quality is concerned with quality of information output, whose measures include the following: perceived usefulness of report-items [9]; accuracy, relevance, and timeliness of information [40]; quality of output and usefulness of system [43].

Having reviewed previous research on software quality, we now derive the software quality attributes that are used for the present

3

research. Our justification for the selection of attributes for software quality is primarily based on the ISO 9126 standard. There are six major characteristics of software products that are specified in the ISO/IEC 9126 quality model [29]: functionality, reliability, usability, efficiency, maintainability, and portability. Usability is associated with the "ease with which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component" [54]. Based on this definition, we substitute usability attribute with "ease of use" attribute (i.e., ease with which a system can be used). ISO 9126 standard defines functionality as a set of attributes that are related to system features that comply with user requirement specification and provide functions that are most suitable to the system users [35]. A system with rich functionality will be most useful to the system users as it provides information that is suitable (relevant) to the users' requirements. As the functionality is the most important quality attribute, we made it more explicit in our research by substituting functionality with relevance and usefulness attributes. We do not include portability as one of the attributes in our research, since it is applicable only in software products that need to be implemented on multiple platforms [28]. Furthermore, efficiency attribute is an internal quality attribute as it deals with time behavior and computing resources consumed; thus, we do not include in our chosen list of software quality attributes. Thus, we arrive at five measures for software quality: reliability, ease of use, maintainability, usefulness, and relevance.

The definitions of the software quality attributes we use in this research follow. Reliability of an information system reflects the frequency of error occurrences in the system. Maintainability of system indicates the efforts needed to correct errors or to make enhancements to the system. Ease of use represents how comfortable users feel in using the system. Usefulness is the extent to which the users may find the system outputs useful in their jobs. Relevance denotes whether the system is used as a part in user's decision making process. These attributes are also used by previous researchers to represent software quality. Our selection of quality attributes in this research is consistent also with the traditional software quality definition given by Dromey [21]: correctness, reliability, usability, and maintainability. Correctness is conformance to user requirements and thus corresponds to the attributes "usefulness" and "relevance" in our research.

2.2. Determinants of software quality

While the perceptions of software practitioners vary considerably [48], previous researchers have analyzed factors influencing success of software projects. For example, Ein-Dor and Segev [23] studied organizational factors affecting success of systems. Boehm [13] pointed out the sources of failed software projects as lack of user involvement, incomplete requirements, and lack of executive support. Other factors that influence project failure/success include users' resistance, lack of skills of IT staff [6], technological changes [50], active risk management, and postmortem reviews [59]. Dyba found that organizational factors are at least as important as technical factors for success in software process improvement [22]. Gopal et al. [25] found through a survey that external institutions such as customers and competitors, and internal institutions such as managers influence success of adoption of software metrics program. The key factors that influence system development methodology in IS projects include functional management involvement, external support and use of models [51].

Technological variables that influence software projects were identified through critical success factors. These factors include network reliability, data communications, I/O services, documentation, hardware repair, and proactive planning [55]. Optimal software system structures [8] and fault avoidance [56] were found to be important strategies for system reliability. Many organizations that used Computer Aided Software Engineering (CASE) tools, structured system analysis and design methods, object-oriented methodology, and walkthroughs during system development had reported dramatic reductions in the number of errors that go undetected [52,62]. Selection of design methodology is considered a factor in improving maintainability of object-oriented software [15].

While some researches addressed 'system failure' from engineering perspective without considering organizational or individual characteristics, other researchers ignored technical factors. Furthermore, most previous research have dealt with predictors of IS project success; but, there has been relatively less attention given to determinants of software quality per say. In this research, we hypothesize that the predictors of IS project success can also be considered as predictors of software quality because the software quality in itself is a predictor of IS success. For example, user involvement is a strong predictor of IS project success [13]. We can assume that user involvement will also influence the software quality. This is because inadequate user involvement will result low software quality because of reduced usefulness of the system for users. Based on previous research, we arrive at a comprehensive list of organizational, individual, and technological variables that may influence software quality (Table 1).

As can be seen from Table 1, the organizational variables pertain to top management, CIOs, turnover of personnel, and the user behavior. These variables are inter-related. For example, higher turnover of users can give rise to more change requests to the system, which may require higher IS budget in the future. Higher top management support and higher rank of CIO within the organizational hierarchy can lead to bigger IS budget. The individual variables pertain to user competencies and their involvement in system development. User training and their knowledge gained through experience will make them more competitive. Furthermore, the higher the level of user involvement in system development, the lower will be their resistance to change due to new IS. The technological variables pertain mostly to the system development environment (database management system (DBMS), programming language, and system development method) and experience and skill levels of IS staff. These two categories of the variables are highly related. The competence of the IS staff (experience and skill) depends on the system development environment variables such as DBMS and programming.

Table 1
Organizational, individual, and technological variables.

Type of variable	Variable
Organizational	Top management support
	Rank of IS manager
	Experience of IS manager
	IS budget
	Number of people in system development
	Turnover in IS department
	Quality of documentation
	Frequency of users' change requests
	Turnover in user groups
	Number of people in company
Individual	Level of user involvement
	User resistance to change
	User competency
	User knowledge of systems
	User training in systems
Technological	Experience of IS staff
	Skill level of IS staff
	Support from IS department
	Type of development method used
	Suitability of development method
	Type of programming language
	Suitability of programming language
	Type of database model used
	Suitability of database model

3. Research model

Our research framework is presented in Fig. 1, where in the relationship between the organizational, individual, and technological factors and software quality is shown. The software quality is represented by five attributes: "reliability", "maintainability", "ease of use", "relevance", and "usefulness". The relationships between independent variables (individual, technical, and organizational factors) and dependent variables (software quality attributes) in the proposed model can be hypothesized as follows.

Technological factors such as system development methodology [56], type of programming language used [31], and type of database used [16] will have direct effect on software quality. Use of modern methodologies for software development will most likely result in error-free systems and easily maintainable systems. Use of an appropriate or suitable programming language facilitates the design of good Graphical User Interface (GUI) design which will result in user friendly or easy-to-use systems. *Thus, we hypothesize that the technological factors are associated with the software quality attributes.*

Individual factors, such as, competence of users [61], training of users [30], users' resistance to change, and user involvement [3] will influence software quality of IS projects. User involvement during IS project development will lead to more meaningful IS outputs and those of relevance to their decision making tasks, since users' input and preferences would have been taken into account. Similarly, trained and competent users will be able to understand IS outputs well and more likely find the IS outputs more useful for their day-to-day tasks. Users' negative attitude towards system implementation more likely will lead to unreliable systems, since such users will request for more frequent modifications of the system. *Thus, we hypothesize that the individual factors are associated with the software quality attributes.*

Organizational factors at the firm level and IS department level [23,30] will have influence on software quality attributes. Good management attitude and the adequacy of IS budget will lead to user friendly systems. The higher the turnover of employees as end users, the more frequent the requests for design changes will occur; this leads to less reliable systems. Higher rank of IS manager will facilitate better understanding of organizational needs, thus leading to more relevant information systems. *Hence, we hypothesize that the organizational factors are associated with the software quality attributes.*

4. Research method

Empirical verification of our model was undertaken using a mail questionnaire. The respondents were asked to answer the questions considering 'a specific' information system application devel-

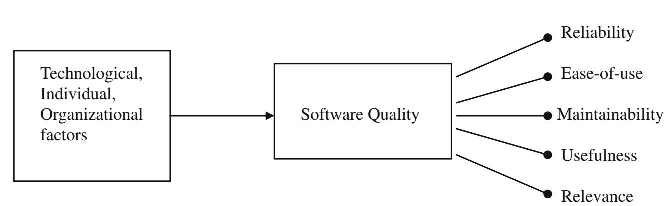

Fig. 1. The research model.

Table 2
Sample items of questionnaire and the scales.

Item	Lickert-type scale	
	1	5
Independent variables		
Level of user training	Excellent	Poor
Suitability of language	Very suitable	Not at all uitable
User resistance to change	High	Low
Level of user competency	High	Low
Frequency of change requests	Frequent	Never
Degree of user involvement	High	Poor
Quality of documentation	Excellent	Poor
Level of support from IS department	High	Low
Errors in the system	Frequent	Never
Ease of use	Very easy	Very difficult
Dependent variables		
System meets user expectation	Very well	Very poor
System usage frequency	Never	Everyday
Relevance of system to decision making	High	Low
Outcome of system	Fail	Successful

Table 3
Characteristics of respondents.

Characteristic	Frequency	Percent
Experience in IS (years)		
<1	4	3.6
1–3	2	1.8
3–5	9	8.0
5–7	5	4.5
>7	92	81.1
Number of people in the company		
<50	6	5.4
50–100	17	15.2
101–500	45	40.2
501–1000	15	13.4
>5000	29	25.9
Number of people involved in developing the system		
1–3	28	25.0
3–5	37	33.0
6–10	18	16.1
11–15	13	11.6
>15	16	14.3
Language used in the system		
COBOL	35	32.14
C/C++	6	5.36
FORTRAN	1	0.89
Application package	26	23.21
Other	40	35.71
Missing data	4	2.69
Database model used in the system		
Relational database	71	63.4
Network database	11	9.8
Distributed database	3	2.7
OO database	8	7.1
Other	19	17.0

oped in their organizations. Table 2 describes sample format of questions used in our survey. Measures were phrased as questions on a five-point Likert-type scale, with anchors at the ends either with the terms "Excellent" and "Poor", "High" and "Low" or "Frequent" and "Never." Target respondents were information system executives from various firms in the USA. This is in accordance with previous practice, where most studies use IS executives as the respondents [17]. The IS executives are most informed of the technical, user, and organizational environment of their system development projects and they are the single source who can answer questions from several directions. We randomly selected 600 companies from the Directory of Top Computer Executives published by Applied Computer Research, Inc. and mailed them the questionnaire. A total of 127 respondents filled and returned the questionnaire. An additional 52 questionnaires were returned by the post office because of incorrect or incomplete address. Out of the 127 questionnaires, 15 were discarded because many fields in these questionnaires were incomplete; thus, there were 112 cases available for analysis. The effective response rate is 127/(600 − 52) = 23%.

Factor analysis was used to find out the underlying dimensions of the determinants of software quality. Factor analysis is a widely used data reduction and summarization technique [14]. In this research, we also use logistic regression analysis [39] to find significant factors in predicting software quality of IS project. Logistic regression, a popular method in social science research [42], has been used by previous researchers [5,17,46] for analysis of data. Logistic regression has the advantages that it requires far fewer assumptions than discriminant analysis. In addition, slight errors in user responses are absorbed by recoding them as binary, thus reducing the impact of errors on the results. We coded the IS projects as reliable/unreliable, easy-to-maintain/hard-to-maintain, easy-to-use/hard-to-use, very-useful/not-useful, and relevant-output oriented/irrelevant-output oriented. We used logistic regression to find the dimensions of determinants/critical factors contributing to software quality measures. The characteristics of the respondents and their firms are given in Table 3.

5. Analysis and results

5.1. Dimensions of determinants of software quality

An objective of our research was to find the dimensions of the determinants affecting software quality of IS projects. Principal component factor analysis was performed on the independent variables using the SPSS procedure FACTOR to enhance construct validity [33] and to dimension the determinants. Analysis using a varimax orthogonal rotation (rotation converged in nine iterations) yielded six factors with eigenvalues greater than 1.0. During factor analysis, 17 items were retained and seven items were dropped (Table 4). In selecting items for the final scale, items with factor loadings greater than 0.50 were retained; this is consistent with previous research [34]. The factor loadings ranged from 0.503 to 0.847. The total amount of variance explained by the six factors was 64.43%. The underlying dimensions were interpreted as: (1) user capabilities, (2) attitude of management, (3) stability of organization, (4) suitability of technology, (5) capability of IS department and (6) responsiveness of IS department. Out of these, three are organizational factors, two are technical factors, and one is individual factor.

The first underlying dimension is named as "user capabilities", which constitutes user characteristics: user competency, user training, user knowledge, user involvement, and user resistance to change. The second factor "attitude of management" includes sufficiency of IS budget provided by management, extent of management support for the system, and documentation quality enforced by management. The third factor, "stability of organization", is concerned with turnover of employees in the company and turnover of IS staff in the system department. The fourth factor is "suitability of technology" and it includes the items of suitability of database management system and suitability of programming language. The fifth factor named as "capability of IS department" contains three items: support from

Table 4
Factor analysis of determinants of IS project success.

Factors	Factor loadings
Factor 1: Capabilities of users (I)	(Eigenvalue = 4.40, pct. of var. = 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, pct. of var. = 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, pct. of var. = 8.62%)
Turnover in IS department	0.847
Turnover in the company	0.839
Factor 4: Suitability of technology (T)	(Eigenvalue = 1.24, pct. of var. = 7.27%)
Suitability of database management	0.823
Suitability of programming language	0.637
Factor 5: Capability of IS department (T)	(Eigenvalue = 1.12, pct. of var. = 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, pct. of var. = 6.08%)
Rank of IS director	0.774
Frequency of users' change requests	0.678

(I) – Individual factor; (O) – organizational factor; (T) – technological factor.

IS department, experience of IS staff, and suitability of development method. The sixth factor is "responsiveness of IS department"; it contains items of relative rank of IS director and frequency of user requests.

Furthermore, reliability of each factor was evaluated using Cronbach's alpha [44]. The Cronbach's alpha is a measure of how well the set of items constituting a factor relate to each other to form a single dimension. It is a measure of internal consistency of the items, which reflects on the homogeneity of the items composing the factor. The recommended value for alpha for exploratory research is 0.60 that have been adopted by other researchers [17,34]. The values of alpha coefficients for the factors found in the study ranged from 0.61 to 0.76 (Table 5), which are above the recommended value.

5.2. Association between factors and software quality attributes

The second objective of our research was to determine the strength of associations between the factors and the software quality attributes. Five software quality attributes (reliability, ease of use, maintainability, usefulness, and relevance) are logit-regressed with the six underlying factors. As per Menard [39], the overall adequacy of the logit-regression model can be judged by model χ^2 and R^2_L (proportional reduction in χ^2). He suggested that R^2_L should be preferred over the conventional coefficient of determination R^2 that is used in linear/non-linear regression. Model χ^2 is obtained by subtracting final -2 Log Likelihood (-2LL) from Initial Log Likelihood function -2LL; we computed R^2_L by dividing model χ^2 by Initial -2LL. Higher values of model χ^2 and R^2_L represent better fit of the model and stronger relationship between independent variables and dependent variables.

Table 5
Factors independent variables: internal consistency.

Factor	Number of items	Cronbach's alpha
Capabilities of users	5	0.7217
Attitude of management	3	0.7570
Stability of organization	2	0.7346
Suitability of technology	2	0.6095
Capability of IS department	3	0.6407
Responsiveness of IS department	2	0.6662

Table 6 has the logit regression results for each software quality attribute with the six factors. The model relating system reliability and the factors has a high goodness-of-fit (χ^2 of 18.51, significant at 1% ($p = 0.0051$) and R^2_L of 0.228). One of the factors, namely 'responsiveness of IS department' has significant influence on "system reliability" ($p = 0.004$). The regression model relating "ease of use" with the factors also has a high goodness-of-fit ($\chi^2 = 27.92$, $p = 0.0001$, $R^2_L = 0.218$). Furthermore, two factors ("capabilities of users" and "attitude of management") have significant ($p = 0.0057$ and 0.0492, respectively) impact on the "ease of use". While the overall model relating system maintainability is not significant (χ^2 is 8.8, $p = 0.1850$) and $R^2_L = 0.093$), the factor 'responsiveness of IS department' has a significant influence on maintainability ($p = 0.0243$). The model relating "usefulness" and the factors is significant (χ^2 16.71, $p = 0.0104$ and a high R^2_L value 0.496); the six factors together can explain about 50% of the variance of the dependent variable "usefulness". In addition, two of the factors, namely "capability of IS department" and "responsiveness of IS department" have significant influence on the software quality attribute, "usefulness" ($p = 0.0694$ and 0.0276, respectively). The fifth model relating "relevance" with the factors has a low goodness-of-fit, implying weak relationship; however, the factor "attitude of management" has a significant effect ($p = 0.0606$) on the "relevance" attribute.

6. Discussion

The research has two objectives: (i) to obtain the factors influencing software quality through statistical analysis and (ii) to determine the association between these factors and software quality attributes of IS projects. We used five measures of software quality: reliability, ease of use, maintainability, usefulness, and relevance. We started with 24 items (technological, individual, and organizational) from which 17 significant items were extracted into six dimensions through factor analysis. These six dimensions accounted for over 64% of the variance. Out of the six factors, there were one individual factor ("capabilities of users"), two technological factors ("suitability of technology" and "capabilities of IS department"), and three organizational factors ("attitude of management", "stability of organization", and "responsiveness of IS department").

6.1. Association between factors and software quality attributes

In order to meet the second objective of determining the association between factors and quality attributes, we tested five models, one for each of the software quality attributes, using logit regression. Out of these, three models (reliability, ease of use, and usefulness) had shown high goodness-of-fit, indicating strong association between the factors and each of these software quality attributes. For the other two models (maintainability and relevance), the associations between factors and quality attributes are weak. These models are described below.

Table 6
Determinants of software quality.

Software quality attribute	Model χ^2 (significance)	Model R^2_L	Significant factors (type)	Significance of factors
Reliability	18.51* ($p = 0.005$)	0.228	Responsiveness of IS department (O)	0.004
Ease of use	27.92* ($p = 0.000$)	0.214	Capabilities of users (I)	0.006
			Attitude of management (O)	0.05
Maintainability	8.80 ($p = 0.185$)	0.093	Responsiveness of IS department (O)	0.024
Usefulness	16.71* ($p = 0.010$)	0.496	Capabilities of IS department (T)	0.069
			Responsiveness of IS department (O)	0.027
Relevance	8.64 ($p = 0.195$)	0.07	Attitude of management (O)	0.060

(I) – Individual factor; (O) – organizational factor; (T) – technological factor.
* Model significant at $p \leqslant 0.01$.

6.2. Reliability

5 Software reliability is defined in terms of error proneness of a system. Respondents were asked to assess the frequency with which errors occurred in the system they selected (1 = frequent, 5 = never). The overall model of association between factors and reliability is significant, indicating the set of factors together can predict the system reliability well. Furthermore, one of the factors, namely "Responsiveness of IS department" has shown significant association with system reliability. Considering this factor's constituting items ("Rank of IS director" and "Frequency of user change requests"), one can infer that less frequent changes (this question is coded from "high" to "low") and higher rank of CIO are associated with higher reliability or fewer software errors. Thus in software projects where the requirements are dynamic or changing, the users request for changes very often; the frequent modifications on the system will result in a "messy" system, which will become hard to understand resulting in more errors in the system. More likely, a high ranked CIO establishes system development standards in the organization, follows modern approaches to system development and is able to allocate more resources to system documentation. These initiatives will result in systems that are better designed/ understandable, resulting in fewer errors in the system. Our results are consistent with previous research [37], which relates CIO rank with system success.

6.3. Ease of use

Ease of use reflects the effort put in using the system or the ease with which the system can be understood, learned, used, and is attractive to use [29,54]. Our results show that the overall model of association between the six factors and "ease of use" is strong ($p < .01$). In addition, the two factors "Capabilities of users" and "Attitude of management" are significantly associated with the "ease of use" attribute. The factor "Capabilities of users" is composed of items: users' knowledge in the computerized system, the training users received in the system, user involvement in the system, user resistance for using the system, and technical competency of the users. Users with the desirable competencies are likely to be conversant with technical and operational aspects of the system and will have little difficulty using the systems. Our findings are consistent with those of previous researchers in that user involvement [4] and computer training [18] are positively related with system success. "Attitude of management" factor consists of items related to management support, budget sufficiency, and quality of documentation. An adequate budget will help develop systems with good user friendly interfaces, which will lead to easy-to-use systems. Good documentation, such as user operational manuals, will help users understand the system operations better, thus leading to easy-to-use systems. Previous research regarding small organizations gives some support to our research that top management involvement is related with IS success [18].

6.4. Usefulness

We measure the usefulness of a system as the extent to which the system meets the user expectation, thus making the system outputs useful to the target stakeholders. When the system outputs meet the users' requirement, more likely the users will be benefited by increased effectiveness and efficiency in their day-to-day jobs. Our results show that the model of association between factors and system usefulness shows strong support ($p < .01$) with a predictability of about 50% of the variation in usefulness.

The factors that have significant influence on system usefulness are related mostly to IS department: "Capabilities of IS department" and "Responsiveness of IS department". The factor "Capabilities of IS department" includes extent of experience of system staff and level of support from the system department. The experienced system staff can understand the users' information better and are able to develop and deliver systems that will be more helpful and useful to the users in their day-to-day tasks. Similarly, system department's support in helping the users (for example, by hand holding and/or trouble shooting), will lead to higher usage and benefits to users. Regarding the factor "Responsiveness of IS department", high ranked CIO can visualize well the company's information needs at the organization level, thus directing the department staff to produce information outputs useful to the organization at large.

As explained above, usefulness represents the strongest model of all other attributes, as its R^2_L is 0.49 and the associated significant factors are IS department related. A possible explanation for these results may be attributed to the data set. The data set shows that 81% of the respondents (CIOs) have more than 7 years of IT experience, up-to five members are present in a typical software development team, and systems were developed mostly in COBOL programming and relational database base environments. Furthermore, these systems were developed in-house. The above shows IS department is strong in technical skills and experience, thus IS staff could select the database software and system development method suitably; this may have resulted in strong association between "Capabilities of IS department" and usefulness. As the systems were developed in-house, the "Responsiveness of IS department" in terms of accommodating user requests for changes is very important for developing "useful" systems.

6.5. Relevance

Relevance is a software quality attribute that represents the extent to which the system outputs are used for decision making purpose. Though the overall model for association between factors and relevance is not supported, the "attitude of management" factor has significant influence on the quality attribute "relevance". This factor is composed of three items: sufficiency of budget, quality of documentation, and support from management. Thus, having an adequate budget will enable systems department to develop decision support and management support systems, beyond transaction processing systems; this will increase the proportion of system projects that are relevant for decision making. Support from management will result in allocating more resources to the system, thereby affording to develop sophisticated systems useful for strategic purposes of organization – implying higher "relevance" of system outputs.

6.6. Maintainability

Maintainability reflects effort that was put into a specified system in order to incorporate changes or fix bugs in the system. Though the overall model of association between factors and maintainability is not significant, one factor "Responsiveness of IS department" has shown significant association with system maintainability. Similar to reliability (as also evident from the high correlation between reliability and maintainability from Appendix A), this association implies that lesser the frequency of changes, the higher the maintainability. Making frequent changes to the system will result in systems that are hard-to-maintain over a period of time. A high ranked CIO more likely enforces standards in system development methodologies, which results in systems that are easily understandable requiring less effort to maintain.

7. Conclusion

The primary objective of this research was to find the significant factors that affect key attributes of software quality. Based on the data collected from 112 CIOs regarding IS projects, we arrived at six determinants (factors) that influence software quality: three organizational factors (Attitude of management, Stability of organization, and Responsiveness of IS department), two technological factors (Suitability of technology and Capability of IS department), and one individual factor (Capabilities of users). We built five logistic regression models to assess the relationship between these factors and each of the five software quality attributes. Our results show that 3 of the 5 models (i.e., reliability, ease of use, and usefulness) have high goodness-of-fit, indicating the validity of our research model. In addition, our results show that four out of the six factors have significant influence (individually) on one or more software quality attributes; these are Capability of users, Attitude of management, Capability of IS department, and Responsiveness of IS department. The other two factors (Suitability of technology and Stability of organization) did not show significant impact on software quality attributes. Thus, our results show that organizational factors have higher impact on software quality compared to technical factors.

7.1. Implications to practice

This research has the following implications for practitioners. First, it provides determinants that can influence software quality in organization, which are not addressed adequately in previous research. The determinants of software quality are useful to CIOs and CEOs (Chief Executive Officers) who are responsible for resource allocations and for improving software quality in organizations. Second, CEOs may adopt several strategies to improve software quality, such as allocation of adequate budget to their IS projects, providing strong support and leadership from the top management, and providing computer and system training to all employees in the organization. Third, CIOs may consider strategies such as, developing better liaison with user departments to ensure user involvement in all phases of system development, enforcing documentation and process standards, and hiring experienced and more cooperative IS staff.

7.2. Limitations

Our results may be interpreted with caution because of the following limitations of our research. First, we chose CIOs as respondents for our research. If we had chosen other subjects, such as end users, business executives, or programmers, then the results could be somewhat different. If business executives were chosen as subjects, then most likely the results would have more organizational emphasis. If technical subjects (say, programmers) were chosen, then the research results could highlight more technical aspects as the most important factor. Thus, it is worthwhile to extend the research by repeating this experiment with business executives and technical IS staff and compare the results. Second, another way to improve upon the present research is to use multiple types of respondents – for example, programmers could be used for answering questions on technological factors, end users could be used to answer questions on individual factors, and CEOs could be used to answer questions on organizational factors. Third, our results show that the top six factors explained 64.43% of the total variance. Some previous studies show somewhat higher % variance ranging from 63.9% to 85%. Some examples of previous research are as follows. The 10-factor instrument to measure stress among software professionals accounted for 63.9% of total variance [49]; the 9-critical success factor components explained approximately 85% of variance in influencing software process improvement initiatives [41]; the 7-factor determinants related to data resource management accounted for 68.6% of total variance [32]. The reason for lower % of variance in our research could be the subjective nature of the questionnaire items and inadequate knowledge of the CIOs regarding the attitudes of end users. Future research based using multiple groups of respondents as discussed in the second item above could improve the research results.

7.3. Implications to research

In addition to the research extensions proposed above in the context of limitations, the current research can be extended in the following directions. First, the present study considered some of the most commonly used software quality attributes. The study may be repeated with other quality attributes not considered in this research, such as portability. Second, software quality should be defined based on the context and not all quality attributes are important in all contexts [28]. Future research may consider a subset of software quality attributes that are important in each environment (for example, for in-house software development, outsourced software, and packaged software) and thus determine the factors that influence those quality attributes. Third, in our research we considered only *external* quality attributes of software. Future research may focus on *internal* quality attributes of software (such as, structural complexity, software size [1], logical/physical structures [2,27] and the factors that influence them). Fourth, a longitudinal study may be conducted through surveys before and after the organizations have adapted the above strategies. Such a study can assess the effectiveness of the above strategies in improving software quality.

Appendix A. Correlation matrix for quality attributes

	Maintainability	Reliability	Usefulness	Relevance	Ease of use
Maintainability					
Pearson correlation	1.000	.208	.127	−.055	.307
Sig. (2-tailed)		.028	.181	.566	.001
Reliability					
Pearson correlation	.208	1.000	.335	.051	.202
Sig. (2-tailed)	.028		.000	.595	.033
Usefulness					
Pearson correlation	.127	.335	1.000	.234	.385
Sig. (2-tailed)	.181	.000		.013	.000
Relevance					
Pearson correlation	−.055	.051	.234	1.000	.182
Sig. (2-tailed)	.566	.595	.013		.054
Ease of use					
Pearson correlation	.307	.202	.385	.182	1.000
Sig. (2-tailed)	.001	.033	.000	.054	

References

[1] M. Agarwal, K. Chari, Software effort, quality, and cycle time: a study of CMM level 5 projects, IEEE Transaction on Software Engineering 33 (2007) 145–156.
[2] M. Alshayeb, Empirical investigation of refactoring effect on software quality, Information and Software Technology 51 (2009) 1319–1326.
[3] H. Barki, J. Hartwick, Measuring user participation, user involvement, and user attitude, MIS Quarterly (1994) 59–82.
[4] J.J. Baroudi, M.H. Olson, B. Ives, An empirical study of the impact of user involvement on system usage and user satisfaction, Communications of the ACM 29 (March) (1986) 232–238.
[5] V.R. Basili, W.L. Melo, A Validation of object-oriented design metrics as quality indicators, IEEE Transactions on Software Engineering 22 (October) (1996) 751–761.
[6] J. Beaver, G.A. Schiavone, The effects of development team skill on software product quality, ACM SIGSOFT Software Engineering Notes 31 (3) (2006) 1–5.
[7] S. Belardo, K.R. Karwan, W.A. Wallace, DSS component design through field experimentation: an application to emergency management, in: Proceedings of the Third International Conference on Information Systems, December 1982, pp. 93–108.
[8] O. Berman, N. Ashrafi, Optimization models for reliability of modular software systems, IEEE Transaction on Software Engineering 19 (11) (1993) 1119–1123. Nov.
[9] B.K. Blaylock, L.P. Rees, Cognitive style and the usefulness of information, Decision Sciences 15 (1) (1984) 74–91 (Winter).
[10] J. Boegh, S. Depanfilis, B. Kitchenham, A. Pasquini, A method for software quality planning, control, and evaluation, IEEE Software (March/April) (1999) 69–77.
[11] J. Boegh, A new standard for quality requirement, IEEE Software (March/April) (2008) 57–62.
[12] B. Boehm, Practical Strategies for Developing Large Software Systems, Addison Wesley Longman, 1975.
[13] B. Boehm, Project termination doesn't equal project failure, IEEE Computer (September) (2000) 94–96.
[14] B.L. Bowerman, R.T. O'Connell, Linear Statistical Models: An Applied Approach, PWS-KENT Publishing Company, 1990.
[15] L.C. Briand, C. Bunse, J.W. Daly, A controlled experiment for evaluating quality guidelines on the maintainability of object-oriented designs, IEEE Transactions on Software Engineering 27 (6) (2001) 513–530. June.
[16] M.J. Cerullo, Information systems success factors, Journal of Systems Management 31 (12) (1980) 10–19.
[17] P.Y.K. Chau, K.Y. Tam, Factors affecting the adoption of open systems: an exploratory study, MIS Quarterly (March) (1997) 1–21.
[18] W.H. DeLone, Determinants of success for computer usage in small business, MIS Quarterly 12 (1) (1988) 51–61.
[19] W.H. DeLone, E.R. McLean, Information system success: the quest for the dependent variable, Information System Research (March) (1992) 60–95.
[20] W.H. DeLone, E.R. McLean, The DeLone and McLean model of information system success, Journal of Management Information System 19 (4) (2003) 9–30.
[21] R.G. Dromey, A model of software product quality, IEEE Transactions on Software Engineering (February) (1995) 146–162.
[22] T. Dyba, An empirical investigation of the key factors for success in software process improvement, IEEE Transactions on Software Engineering 31 (5) (2005) 410–424.
[23] P. Ein-Dor, E. Segev, Organizational context and the success of management information systems, Management Science 24 (June) (1978) 1064–1077.
[24] C.R. Franz, D. Robey, Organizational context, user involvement and the usefulness of information systems, Decision Sciences 17 (3) (1986) 329–356 (Summer).
[25] A. Gopal, T. Mukhopadhyay, M.S. Krishnan, The impact of institutional forces on software metrics programs, IEEE Transaction on Software Engineering 21 (8) (2005) 679–694.
[26] N. Gorla, Techniques for application software maintenance, Information and Software Technology 33 (1) (1991) 65–73.
[27] N. Gorla, R. Ramakrishnan, Effect of software structure attributes on software development productivity, The Journal of Systems and Software 36 (2) (1997) 191–199.
[28] R.L. Glass, Defining Quality Intuitively, IEEE Software (May/June) (1998) 103–107.
[29] ISO/IEC 9126, Software Engineering – Product Quality – Part 1: Quality Model. International Standards Organization, Geneva, Switzerland, 2001.
[30] B. Ives, S. Hamilton, G.B. Davis, A framework for research in computer-based management information systems, Management Science 26 (9) (1980) 910–934.
[31] B. Ives, M.H. Olson, J.J. Baroudi, The measurement of user information satisfaction, Communications of the ACM 26 (10) (1983) 785–793.
[32] H. Jain, K. Ramamurthy, H.-S. Ryu, M. Yasai-Ardekani, Success of data resource management in distributed environments: an empirical investigation, MIS Quarterly (March) (1998) 1–29.
[33] F.N. Kerlinger, Foundations of Behavioural Research, Holt, Rinehart, and Winston, New York, 1973.
[34] W.R. King, T.S.H. Teo, Key dimensions of facilitators and inhibitors for the strategic use of information technology, Journal of MIS 12 (4) (1996) 35–53 (Spring).
[35] B. Kitchenham, S.L. Pfleeger, Software quality: the elusive target, IEEE Software 13 (1) (1996) 12–21.
[36] S. Lauesen, H. Younessi, Is software quality visible in the code?, IEEE Software (July/August) (1998) 69–73
[37] H.C. Lucas Jr., Performance and the usage of an information system, Management Science 21 (1975) 908–919.
[38] J. Luftman, R. Kempaiah, Key issues for IT executives 2007, MIS Quarterly Executive 7 (2008) 99–112.
[39] S. Menard, Applied Logistic Regression Analysis, Sage Publications, Newbury Park, CA, 1995.
[40] J. Miller, B.A. Doyle, Measuring effectiveness of computer based information systems in the financial services sector, MIS Quarterly 11 (1) (1987) 107–124.
[41] M. Montoni, A.R. Rocha, A Methodology for Indentifying Critical Success Factors that Influence Software Process Improvement Initiatives: An Application in the Brazilian Software Industry, Lecture Notes in Computer Science LNCS 4764 (2007) pp. 175–186.
[42] P.S. Morgan, D. Techman, Logistic regression: description, examples, and comparison, Journal of Marriage and the Family 50 (1988) 929–936.
[43] A.I. Nicolaou. An Empirical Examination of Information Systems Success in Relation with Information Systems Development Phenomena. D.B.A. Dissertation, Southern Illinois University at Carbondale, 1993.
[44] J. Nunally, Psychometric Theory, McGraw-Hill, New York, 1967.
[45] J. Offutt, Quality attributes of web software applications, IEEE Software (March/April) (2002) 25–32.

[46] H.M. Olague, L.H. Etzkorn, S. Gholston, S. Quattlebaum, Empirical validation of three software metrics suites to predict fault-proneness of object-oriented classes developed using highly iterative or agile software development processes, IEEE Transactions on Software Engineering 33 (6) (2007) 402–419.
[47] D.L. Parnas, The role of inspection in software quality assurance, IEEE Transactions on Software Engineering 29 (8) (2008) 674–676.
[48] J. Pereira, N. Cerpa, J. Verner, M. Rivas, J.D. Procaccino, What do software practitioners really think about project success, The Journal of Systems and Software 81 (2008) 897–907.
[49] K.S. Rajeswari, R.N. Anantharaman, Development of an instrument to measure stress among software professionals: factor analytic study, in: SIGMIS Conference, April 2003, pp. 34–43.
[50] J.S. Reel, Critical success factors in software projects, IEEE Software (May/June) (1999) 18–23.
[51] T.L. Roberts Jr., M.L. Gibson, K.T. Fields, R.K. Rainer Jr., Factors that impact implementing a system development methodology, IEEE Transactions on Software Engineering 24 (8) (1998) 640–649.
[52] M. Rumbaugh et al., Object-Oriented Modeling and Design, Prentice-Hall, Inc., 1991.
[53] P.B. Seddon, A respecification and extension of the DeLone and McLean model of IS success, Information Systems Research 8 (3) (1997) 240–253.
[54] A. Seffah, E. Metzker, The obstacles and myths of usability and software engineering, Communications of the ACM 47 (12) (2004) 71–76.
[55] D.P. Slevin, Critical success factor analysis for information systems performance measurement and enhancement, Information & Management 21 (3) (1991) 161–174.
[56] I. Sommerville, Software Engineering, third ed., Addison-Wesley Publishing Company, 1995.
[57] E.B. Swanson, Management information systems: appreciation and involvement, Management Science 21 (2) (1974) 178–188.
[58] J. Tian, Quality-evaluation models and measurements, IEEE Software (May/June) (2004) 84–91.
[59] J.M. Verner, W.M. Evanco, In-house software development: what project management practices lead to success?, IEEE Software (January/February) (2005) 86–93
[60] J. Voas, W.W. Agresti, Software quality from a behavioral perspective, IT Professional (July/August) (2004) 46–50.
[61] K.B. White, Information systems development success: perspectives from project team participants, MIS Quarterly 10 (3) (1986) 214–223.
[62] E. Yourdon, Modern Structured Analysis, Prentice-Hall, Inc., 1989.