International Journal of Computing Sciences Research (ISSN print: 2546-0552; ISSN online: 2546-115X)
Vol. 1, No. 2, pp. 46-65
doi: 10.25147/ijcsr.2017.001.1.12
https://stepacademic.net

Long Paper

Software Quality: Assessment on the Organizational, Technological and User-related Determinants in the Philippine Setting

Jerhmie James A. Zacarias
The Graduate School, University of Santo Tomas
jerhmie@gmail.com

Date received: July 31, 2017
Date received in revised form: October 18, 2017
Date accepted: October 20, 2017

Recommended citation:

Zacarias, J. J. A. (2017). Software quality: Assessment on the organizational, technological and user-related determinants in the Philippine setting. *International Journal of Computing Sciences Research*, *1*(2), 46-65. doi: 10.25147/ijcsr.2017.001.1.12.

Abstract

Purpose – This mixed method descriptive study aimed to show the impact of Organizational, Technological and User-related factors on software quality. It also assessed the strength of association between these factors and software quality attributes.

Method – Toward this goal, data from 405 employees of member companies of the Philippines Software Industry Association (PSIA), together with Emerson Electric Asia and NGA Human Resources were analyzed.

Results and Conclusion – The results showed that values of sampling adequacy Kaiser-Meyer-Olkin (KMO) for organizational, technological and user-related determinants are higher than the standard, indicating a good and reliable sample size. The exploratory factors were selected through series of Monte Carlo PCA for parallel analysis on which three components were extracted and retained from each group of determinants that best represented the factors. Organizational determinants of software quality retained “Company Culture”, “Time Resource” and “Process Management”. User-related determinants retained “Level of User Involvement”, “User Resistance to Change” and “Users’ Training on Systems”. Technological determinants retained “Input-Output Services”, “Type of Database Model Used” and “Network Reliability”. Results of logistic regression then showed that organizational, user-related and technological determinants are strongly associated with software quality presented in five attributes: Ease of Use, Maintainability, Relevance, Reliability and Usefulness.

Research Implications – This study is by far the first in the Philippines and is very relevant as the country is slowly becoming a global titan in the Business Process Outsource business. The results obtained are beneficial for the academe to devise curricula that would prepare students to the IT industry, and to industry’s senior management to devise software quality improvement programs to align IT strategies to business strategies; thus, having an IT enabled business strategy.

Keywords – IT enabled business strategy, Kaiser-Meyer-Olkin, logistic regression, Monte Carlo Principal Component Analysis, software quality

INTRODUCTION

The increasing dependence of organizations on information systems and the increasing losses to organizations due to poor software, compel management’s attention towards the improvement of software quality. Many Information System professionals when asked about what they understand by software quality immediately start to talk about functional testing, on which a specific action or function of the code is being tested on how well it complies or conforms to a given software design based on the functional requirements or specifications of the system. It should be noted that software quality is not only about technical aspects such as the source code or programming syntax, but also includes what user sees. Software quality is more than static assessment; it includes non-functional or behavioral attributes like reliability and maintainability. 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 since the advent of information technology and though its importance is clearly seen as an integral part of information system success, it seems that there is a need for more research to investigate the causes of poor software quality. There have been few comprehensive studies and literature on factors that influence software quality, and most studies addressed technical factors of software quality without adequately considering the user and organizational influence on software quality. Also, quantitative survey-based research, combined with expert opinions regarding software quality, is lacking. Lastly, no research about this topic on the Philippine setting can be found.

This study aimed to determine the external factors that influence software quality. These factors could be organizational, technological or end-user related. It also aimed to assess the strength of association between these factors and software quality attributes.

This study has a big implication on Information Technology (IT) practitioners because it can present the determinants that can affect software quality in organizations which were not addressed adequately in previous studies. The results of this study are beneficial for the Chief Information Officers (CIO) to devise software quality improvement programs to align IT strategies to business strategies. Through this study, CIOs may adopt several strategies to improve software quality such as allocating adequate budget for software projects and providing support and leadership from top management. Furthermore, based on a study from the Philippine Software Industry Association (2010), from a sourcing perspective, the Philippines is one of the global titans. In 2010, IT and BPO revenues totaled $9 billion in the Philippines and its global market share of the offshore services market rose from 7% to 8%. Therefore, the result of this study could give facts that can be used by IT professionals to improve software quality, thus helping the Philippines maintain its competitive edge.

LITERATURE REVIEW

Software Quality

This study was based on the study made by Gorla and Lin (2010), which is about the determinants of software quality. Wang, Khoshgoftaar, Hulse, and Gao (2011) stated that real-world software systems are becoming larger, more complex, and much more unpredictable. Software systems face many risks in their life cycles. Software practitioners strive to improve software quality. Gorla and Lin (2010) arrived at six determinant or factors that influenced 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). They built five logistic regression models to assess the relationship between these factors and each of the five software quality attributes. The results of their study showed that three of the five models which are reliability, ease of use, and usefulness have high goodness-of-fit, indicating the validity of his research model. In addition, the results showed that four out of the six factors have significant influence 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, the results showed that Organizational factors have higher impact on software quality compared to technical factors.

In terms of organization's behavior effect on software quality, Ambrose and Chiravuri (2010) concluded on their study that organizational downsizing does not always realize its strategic intent and has detrimental impact on organizational performance. Their study extended the notion that downsizing negatively impacts performance and that organization downsizing can potentially be detrimental to software quality performance.

Company culture has an effect to the whole organization and every business aspect; including software quality on IT organizations. Aktas, Cicek and Kiyak (2011), Wilderom, Den Berg, and Wiersma (2012), Zehir, Ertosun, Zehir, and Müceldili (2011) all agreed on this. They stated that company culture exists in many versions such as shared philosophy, ideology, value, assumption, beliefs, hope, behavior and norms that bound the organization together. True diagnosis of organization culture, as well as determination of strategy, politics, and human resource will enable an organization to reach its desired organizational efficiency.

Proper process management has also been suggested as an important organizational factor. Barney, Mohankumar, Chatzipetrou, Aurum, and Wohlina (2013) showed that cultural differences and misalignment, and bad quality of process management are the common elements for poor quality of software. They emphasized the need to align company internal success-critical stakeholder groups in their understanding of quality in software development. Trkman (2010) stated that process management, continuous improvement and proper fit between business process tasks and information systems must exist for these would increase the likelihood of continuous success of software development projects.

On technological and organization factors, Ezamly and Hussin (2011) stated that regardless of how much effort you make for the success of software projects, many software projects have very high failure rates and risks during their life. In the advent of cloud computing, another factor deemed to be considered as a technological determinant is network reliability. McKendrick (2013) found out that among 572 business leaders surveyed, almost three-fourths indicated their companies have piloted, adopted or substantially implemented cloud in their organizations. Another determinant deemed important is law. Bush (2007) said in his study that quality can be improved if the computer industry will have regulations and standards. This control can take the form of statutes, rules, and liability.

Software Development and its Relationship to Quality

Software development is the activity or process involved in the development of software products; thus, it has a direct effect on software quality. Rothenberger, Kao, and Wassenhove (2010) stated that software development is similar to other engineering disciplines on which a managed approach is likely to improve the quality of the product. Software development does not end in coding. It goes through implementing and part of implementing is deciding on how the users would be trained. Sarngadharan (2010) said that user training on systems is important because the absence of proper training to use an information system renders the information system obsolete. Training and development help in increasing the job knowledge and skills of employees at each level. That, in turn, helps the organization further achieve its long-term goal.

According to Vavpotic and Bajec (2009), a software development methodology can be defined as a recommended means to achieve the development of program systems, based on a set of rationales and an underlying philosophy. Hauge, Ayala and Conradi (2010) added that there is no set of development practices that is universal to all software development projects.

Scarpa and Puliafito (2009) stated that to develop "high quality software", it is necessary to follow a methodology that characterizes the software development project, with the final goal to optimize time needed and resources required. Database design must also be considered according to Yaldex.com (2013), as it is crucial for a high performance application.

Human Factors Effect on Software Quality

Hazzan and Hadar (2008) proposed that software engineering community should aim to build a measurement model for tracking of software processes, which will relate to technical, managerial and human aspects of such processes. Their study claimed that software intangibility can largely explain why many of the problems associated with software engineering is human related.

Wong and Tate (1994) stated that Level of User Involvement in information system development is considered to be an important factor influencing implementation success or failure. Kujala (2003) also supported the claim when he stated that user involvement is a widely accepted principle in development of usable systems. Li, Chang, Chen, and Jiang (2010) stated that the ability to respond to changes in the environment during the development of software is crucial in achieving a quality product.

Synthesis of the Study

Organizations, together with business leaders, often view software quality as a luxury or cost. It is something that can be sacrificed, if necessary, for more functionality, faster development, or lower costs. However, organizations that have a firm commitment to quality can actually speed development, reduce costs, and add new features with greater ease. Software that has quality will have a value to a department which would then have an immediate outcome of a valuable department to an organization, and an impact of valuable information systems to an organization and economy.

Though software quality problems have been acknowledged through the years, the fact that software quality is of great concern in organizations even in the present implies that there is a need for more research focused on the causes behind poor software quality. It has been seen that most of the previous studies and literature reviewed focused on the technical aspect such as testing of source code or programming syntax but software quality is more than static assessment; it includes non-functional attributes such as reliability and maintainability.

Studies showed that software quality is being influenced from the outside-in by organizational, technological, and individual or end-user related factors. It is also influenced by the current software ecosystem, software functional size, current global economy, and software development process methodology, human factors in development, design, and change management from previous to the proposed system.

Based on the studies reviewed, the researcher arrived at a comprehensive list of organizational, individual, and technological variables that may influence software quality. Majority came from the study of Gorla and Lin (2010), Nagadevara (2008), and Ezamly and Hussin (2011); however, some of their variables were omitted by the researcher because of redundancy and scope of the study.

Furthermore, this study adopted the software quality attributes stated by Gorla and Lin (2010). These are Reliability, Ease of Use, Maintainability, Usefulness, and Relevance. They stated that ease of use is associated with the ease in which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component. They added that functionality is a set of attributes related to system features that comply with user requirement specification and provide functions which are most suitable to the system users. A system with rich functionality will be most useful to the system users as it provides information that is relevant to the users' requirements. Because functionality is the most important quality attribute, the researchers made it more explicit in their study by substituting functionality with relevance and usefulness attributes. The five software quality attributes are presented in Table 2.

Lastly, the measurement of quality is addressed. Previous studies suggested the use of factor analysis to find the underlying dimensions of the determinants of software quality and logistic regression analysis to find significant factors in predicting software quality. The researcher followed the suggestions coming from other studies and used the suggested statistical tools in the outcome of this study.

Research Questions

This study was performed to determine and to test the relationship between the organizational, technological and user-related factors and software quality (Table 1). Software Quality was presented in five attributes: Ease of Use, Maintainability, Relevance, Reliability and Usefulness.

Particularly, the study aimed to answer the following questions:

1. What is the influence of organizational factors on software quality of information system projects?
2. What is the influence of technological factors on software quality of information system projects?
3. What is the influence of user-related factors on software quality of Information system projects?

4. What is the strength of association between the determinants of software quality and Ease of Use as a as software quality attribute?
5. What is the strength of association between the determinants of software quality and Maintainability as a as software quality attribute?
6. What is the strength of association between the determinants of software quality and Reliability as a as software quality attribute?
7. What is the strength of association between the determinants of software quality and Relevance as a as software quality attribute?
8. What is the strength of association between the determinants of software quality and Usefulness as a as software quality attribute?

Hypotheses

The following null hypotheses were formulated and tested in the study.

- **Ho1:** Organizational factors do not influence software quality.
- **Ho2:** Technological factors do not influence software quality.
- **Ho3:** User-related factors do not influence software quality.
- **Ho4:** Organizational, Technological and User-related determinants of software quality are not associated with Ease of Use as a software quality attribute.
- **Ho5:** Organizational, Technological and User-related determinants of software quality are not associated with Maintainability as a software quality attribute.
- **Ho6:** Organizational, Technological and User-related determinants of software quality are not associated with Reliability as a software quality attribute.
- **Ho7:** Organizational, Technological and User-related determinants of software quality are not associated with Relevance as a software quality attribute.
- **Ho8:** Organizational, Technological and User-related determinants of software quality are not associated with Usefulness as a software quality attribute.

Research Paradigm

The outcome of this study was guided by the research paradigm presented in Figure 1. The empirical verification of the paradigm was done through the use of survey questionnaire that was sent to the selected software development companies. Three types of variable were used: Organizational, Technological and User-related. Factor Analysis was used to find out the underlying dimensions of the determinants of software quality. Logistic regression analysis was also used to find the significant factors in predicting software quality. The results were categorized through the determinants stipulated on the study made by Gorla and Lin about the determinants of software quality.

METHODOLOGY

Research Design and Meaning

This study made use of mixed quantitative and qualitative approach focusing on descriptive method-correlational. Descriptive research design was used in the study. The respondents to the survey instrument answered the questions freely based on their own perception. The current state of their employment was the precursor used in observations and in testing for the quantitative worth of the variables. Correlational method, a descriptive type of research, was employed since one of the objectives was to assess the strength of association between the factors that influence software quality and software quality attributes. This study applied mixed method, as well as quantitative (through the use of online survey questionnaire from which resulting data has been statistically analyzed and tested) and qualitative measures (through comparing the results with the previous international and related studies). By getting the perception of the software developers, the results were factored and were correlated to the software quality attributes.

Table 1. Independent Variables

Types of Variables	Sub Variables
Organizational	Top Management Support Experience of IS / IT Manager Budget on Information System Number of People Assigned in Systems Development Turnover in IS Department Quality of Documentation Frequency of User or Client's Change Request Number of Employees in the Company Proper Feasibility Study Time Resource Adequacy of Office Space / Working Environment Terms and Conditions on the contract (organization and client) Process Management Company Culture Company Mission and Vision Support from IS/IT Department
User-related	Level of User Involvement User Resistance to Change User's Competency User's Knowledge of Similar Types of System User's Training on Systems User's Communication Skills User's Cultural Awareness Turnover in User Groups
Technological	Years of Experience of Software Developer Skill Level of Software Developer Type of Software Development Suitability of Software Development Method Used Type of Programming Language Used Suitability of Programming Language Used Type of Database Model Used Suitability of Database Model Used Network Reliability Input / Output Services Hardware Repair

Table 2. Dependent Variables (Software Quality Attributes)

Dependent Variables (Software Quality)
1. Ease-of-use
2. Maintainability
3. Relevance
4. Reliability
5. Usefulness

Subjects and Study Site

The researcher chose the respondents from the list coming from the Philippine Software Industry Association (PSIA). Registered members of the PSIA are certified to be on the software industry. Registrations coming from the Securities and Exchange Commission (SEC) and Department of Trade and Industry are required prior to joining PSIA. The researcher also considered Emerson Electric Asia and NorthgateArinso. Emerson Electric Asia was chosen because it is one of the world’s oldest and largest technology companies. As of 2012, it has a workforce of approximately 127,800 employees worldwide, with a global presence spanning 150 countries. NorthgateArinso was also chosen because it is a leading software & services provider offering innovative HR business solutions to employers of all sizes, including Global Fortune 500 companies and many Public Sector organizations.

The respondents were primarily composed of those within the age 25-34 bracket (61%), followed by those within the ages 18-24 bracket (20%) and ages 35-54 (19%). There is an equal percentage on the gender of the respondents. On years of experience, majority of the respondents have 3-4 years of experience at 35%, 1-2 years of experience at 24%, 5-6 years of experience 21%, 6 months to 1 year experience at 11%; and 7 years or more at 9%. On educational attainment 74% of the respondents hold bachelor degrees, 15% hold post graduate degrees, 9% hold associate degrees and only 2% do not hold any degree.

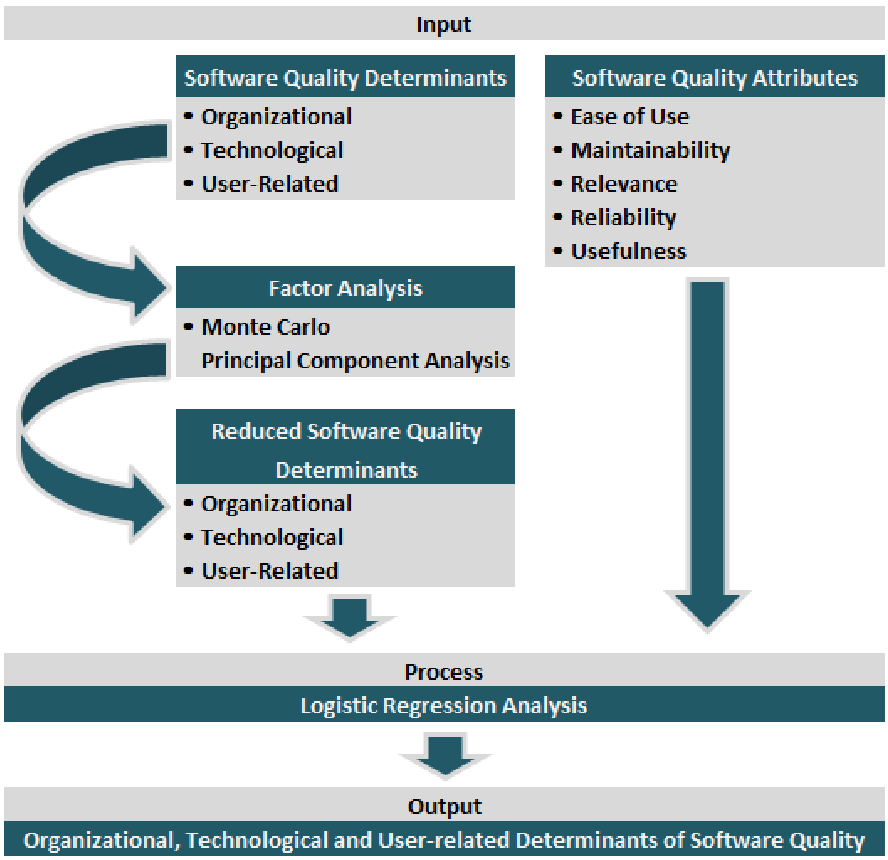

Figure 1. Research Paradigm

Research Instrument and Data Measure

Survey questionnaire was utilized to collect data for this research. The questionnaire that the researcher made was composed of two parts. First was the demographic profile to know the personal information, experience and educational attainment of the respondents. The second part was about software quality and its determinants. The questionnaire was tested using Cronbach's Alpha to make sure that it is reliable and that there is internal consistency.

Data Gathering Procedure

Primary Source

A semi-structured survey questionnaire consisting of in-depth descriptive questions was formulated concerning the factors that influence software quality. A continuum of control was used on the questions. Software developers from the chosen and participating companies were the respondents. Survey was administered online using SurveyGizmo.

Data Analysis

The researcher used the following statistical analysis tools:

- *Descriptive Statistics.* Mean and frequency distribution were measured in this study.
- *Factor Analysis / Principal Component Analysis.* Factor analysis / Principal Component Analysis was performed on the independent variables to find the dimensions of the determinants affecting software quality, thus enhancing construct validity.
- *Logistic Regression.* Logistic regression was used to find the dimensions of determinants/ critical factors contributing to software quality measures.

RESULTS AND DISCUSSION

The result presents the external factors that influenced software quality. These factors could be organizational, technological or end-user related. It also presents the strength of association between these factors and software quality, presented in five attributes: ease of use, maintainability, reliability, relevance, and usefulness. A total of 795 responses were gathered by the researcher however, only 405 passed the data completeness assessment. Responses gathered by the researcher contained demographics showing the age, gender, company name, company size, years of experience in software development, and level of education of the respondents.

Results and Discussion based on Objective 1:

The first objective of this study was to know the external factors that influence software quality. These factors could be Organizational, Technological or User-related. The results showed that values of sampling adequacy by Kaiser-Meyer-Olkin (KMO) for organizational (0.906), user-related (0.858) and technological determinants (0.930) are higher than the standard 0.60 indicating a reliable result. The KMO test revealed that the 405 observations for this study was a good sample size that is reliable and not "mere mathematical illusions" (SPSS Wikipedia, 2003).

The exploratory factors were selected through the series of Monte Carlo PCA for parallel analysis. The factors showed high initial eigenvalues for each respective determinant (organizational, user-related and technological) over the Monte Carlo PCA values for parallel analysis. This indicated that three components have to be extracted from each group of determinants.

Organizational determinants of software quality showed that "**Company Culture**"**,** "**Time Resource**" and "**Process Management**" were retained because they are the three highest loading items that best represent the component or factor. User-related determinants showed that "**Level of User Involvement**"**,** "**User Resistance to Change**" and "**User's training on Systems**" passed the parallel analysis test of the Monte Carlo PCA and represent the next factor. Technological determinants revealed that "**Input-Output Services**", "**Type of Database Model Used**" and "**Network Reliability**" represent the last set of factor. The last component of the software quality model was the demographic factor represented by the following items or indicators: age in years (age), years of experience in software development (years of experience), gender (male = 1, female = 0), and education (associate degree = 1, some college = 0; bachelor of science (bs) = 1, some college = 0; and post graduate (post) = 1, some college = 0).

Cross Validations of the Results to Related Literatures and Studies

On the retained organizational determinants, studies by Aktas et al. (2011), Wilderom et al. (2012), and Zehir et al. (2011) all agreed that, indeed, **company culture** has an effect to the whole organization and has an impact on every business aspect, including software quality in IT organizations. Culture is a model of norms, values, beliefs and attitudes which affects organizational behavior. Company culture exists in many versions such as shared philosophy, ideology, value, assumption, beliefs, hope, behavior and norms that bound the organization together. True diagnosis of organization culture enables an organization to reach its desired organizational efficiency. Results of the study by Barney et al. (2013) showed that cultural differences and misalignment, and bad quality of **Process Management** (which was also a resulting factor on this study) were the common elements for poor quality of software. They emphasized the need to align company internal success-critical stakeholder groups in their understanding of quality in software development. Trkman (2010) stated that process management, continuous improvement and proper fit between business process tasks and information systems must exist for these would increase the likelihood of continuous success of any software development project.

On **Time Resource** as a determinant of software quality, Ambler (2013) stated that when development teams were forced to deliver more functionality than they have time or resources for, they were often motivated to take short cuts which inevitably result in poor quality. On user-related determinants, Wong and Tate (1994) stated that **Level of User Involvement** in information system development was an important factor influencing implementation success or failure. Kujala (2003) also supported the claim when he stated on his study that user involvement was a widely accepted principle in development of usable systems. Moreover, his analysis revealed that user involvement has generally positive effects, especially on user satisfaction and some evidences exist to suggest that taking users as a primary information source was an effective means of requirements capture. This affirmed what Hanssen (2012) mentioned on the related studies section of this study when he said that Software ecosystems is an emerging trend within the software industry, implying a shift from closed organizations and processes towards open structures, where actors external to the software development organization are becoming increasingly involved in development.

Klaus and Blanton (2010) concluded on their study that **User Resistance** was an important issue in the implementation of systems. User resistance must be further understood since it has been found to be at the root of many enterprise software project failures. Resistance to change is important to consider since minor resistance can reduce the speed of change and major resistance can ultimately cause management to abandon its plans (Davidson, 1994).

According to Sarngadharan (2010), the next resulting factor which is **User training on systems** is important because without the proper training to use an information system renders the information system obsolete. Training and development help in increasing the job knowledge and skills of employees. It helps to expand the horizons of human intellect and the overall personality of the employees. It is particularly important to train users of an information system because it helps in increasing the productivity of the user. That in turn helps the organization further to achieve its long-term goal.

On technological determinants obtained in this study, according to Yaldex.com (2013), a good **database design and model** are crucial for a high performance application. Without optimized relationships, a database will not perform as efficiently as possible. This was made firm by Rob and Coronel (2009). According to them, the importance of data modeling cannot be overstated. Data constitute the most basic information units employed by a system. Applications were created to manage data and to help transform data into information. They further iterated that when a good database is available, it does not matter that an applications programmer's view of the data is different from that of the manager and/or the end user. Conversely, when a good database is not available, problems are likely to ensue.

Another resulting factor is **Input and Output Services**. This could be related to the publication by Princeton University (2013) about garbage in and garbage out. According to it, computers will unquestioningly process the most nonsensical of input data (garbage in) and produce nonsensical output (garbage out). Input devices are used to enter data in the computer, thus, if the input device is of bad condition, this can affect the data that is received by the computer. The same is true on output device. If the monitor, for example, shows incorrect data, that will have a detrimental effect on the software because it will not show its intended output.

Network Reliability is also a resulting factor. Computer network or data network is a telecommunications network that allows computers to exchange data. The connections between networked computing devices (network nodes) are established using either through cable media or wireless media. The best-known computer network is the Internet.

Hambling and Goethem (2013), and Cimperman (2006) stated that there are different levels of testing such as unit testing, system testing, end-to-end testing, and User Acceptance Testing. Majority of these requires reliable network connection. Cimperman (2006) stated that the goal of user acceptance testing is to validate that a system of products is of sufficient quality to be accepted by the users, and ultimately, the sponsors. In the current BPO era, this happens when certain programs developed here in the Philippines are being sent to the clients offshore for them to test if the programs conform to their requirements. Since development and UAT are being done on different places, this again requires reliable network connection.

Taking again into consideration the Software Development Life Cycle, another phase deemed to be requiring network reliability is on the implementation phase. There are different types of systems and some of them require to be deployed over the internet or cloud. Some typical types of programs that require network connection are the dynamic websites and cloud computing programs. Shoemaker (2009) said that dynamic website development typically involves server side coding, client side coding and database technology. The other type of program that requires internet connection is cloud-computing based program. On cloud computing, McKendrick (2013), on his study with the Economist Intelligence Unit and IBM found out that among 572 business leaders surveyed, almost three-fourths indicate their companies have piloted, adopted or substantially implemented cloud in their organizations.

Results and Discussion Based on Objective 2:

This study aimed to assess the strength of association between the Organizational, Technological and User-related factors, and software quality attributes. This objective is also related to research questions 4-8. The dependent variables are ease of use (E), maintainability (M), reliability (R1), relevance (R2) and usefulness (U) of software quality. These software quality attributes were measured in binary or dichotomous values that were answered in nominal term (true = 1, false = 0) in the questionnaire.

The computed percentage of probability of software quality with true (labeled as 1) ease of use, maintainability, reliability, relevance, and usefulness is 49.6%, 63.9%, 54.2%, 42.1% and 54.3%, respectively. The probability of a software quality is false (labeled as 0) is estimated to be 50.4%, 36.1%, 45.8%, 57.9% and 45.7%, respectively. Average probability of finding a software quality with true (=1) attributes (ease of use, maintainability, reliability, relevance and usefulness) is 52.8% and 47.2% for a false (= 0) attributes. Since the dependent variables (software quality) are qualitative or dummy, Gujarati (1999) discussed that the Logit model can be used in this case. This method was also used in related studies.

The filtered or reduced forms of the software quality are expressed as follows:

$$Y_i = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_4x_4 + b_5x_5 + b_6x_6 + b_7x_7 + b_8x_8 + b_9x_9 + e_i$$

where:

Y_i = attributes of software quality in natural logarithm, $i = 1, 2, 3, 4, 5$
Y_1 = ease of use, Ln (1/(1-P)
Y_2 = maintainability, Ln (1/(1-P)
Y_3 = reliability, Ln (1/(1-P)
Y_4 = relevance, Ln (1/(1-P)
Y_5 = usefulness, Ln (1/(1-P)
x_1 = factor for organizational determinants (culture-resource-management)
x_2 = factor for user-related determinants (user reaction)
x_3 = factor for technological determinants (services & reliability)
x_4 = age in years
x_5 = years of experience in software development
x_6 = gender: male = 1, female = 0
x_7 = level of education: associate degree (ad) = 1, some college = 0
x_8 = level of education: bachelor of science (bs) = 1, some college = 0
x_9 = level of education: post graduate (post) = 1, some college = 0
e_i = error term
b_i = parameters tested at 5% level, $i = 1, 2, 3, 4, 5, 6, 7, 8, 9$

The Excel and SPSS version 17 facilitated the consolidation and stepwise log-linear computations for 405 observations. The results follow the general problem that software qualities (ease of use (E), maintainability (M), reliability (R1), relevance (R2), and usefulness (U) of software quality) are affected by organizational, user-related and technological determinants. Hence, the presentation of the result is per software quality versus the determinants.

Ease of Use versus Organizational, User-related and Technological Determinants

The ease of use equation can be expressed as:

Y_1 = ease of use, $\text{Ln}\,(1/(1\text{-}P) = b_0 + b_1x_1 + b_3x_3 + b_4x_4 + b_5x_5 + b_8x_8$

where:

x_1 = factor for organizational determinants (culture-resource-management)
x_3 = factor for technological determinants (services & reliability)
x_4 = age in years
x_5 = years of experience in software development
x_8 = level of education: bachelor of science (bs) = 1, some college = 0

Table 3 shows the logit estimates for the determinants of software quality. The log of the odds ratio (logit) in favor of finding a "true" (labeled as 1) answer on ease of use software quality increase by 0.09%, 0.10%, and 0.01% for every 100% increase in factor 1 (culture-resources-management or organizational determinants namely company culture, time resource and process management), factor 2 (service & reliability or technological determinants: input-output services, type of database model used and network reliability), and age (in years, respectively. The standardized beta-coefficients implies that technological determinants (services and reliability = 0.1620), organizational determinants (culture-resources-management = 0.1415) and age (in years = 13.83) explain the odds of finding "true" ease of use of software quality. This indicates that technological - organizational determinants and maturity of individuals help in providing higher software quality in terms of ease of use at 5% level.

Table 3. Logit Estimates for the Ease of use of Software Quality

Model	Parameters	Unstandardized Coefficients (B)	Standardized Coefficients (Beta)	t	Sig.	avg	bs = 1	some deg = 0
5	(constant)	0.68		214.5 8	0.00	0.68	0.68	0.68
b1	cultural-resource-mgt	0.00	0.14	2.61	0.01	8.26	0.01	0.01
b8	bs=1, some degree =0	0.00	-0.18	-3.86	0.00	1.00	0.00	0.00
b3	service reliability	0.00	0.16	3.01	0.00	8.32	0.01	0.01
b5	years of experience	0.00	-0.15	-2.93	0.00	3.47	0.00	0.00
b4	age in years	0.00	0.14	2.63	0.01	30.59	0.00	0.00
a	Dependent Variable: ease of use						0.69	0.70
Model Summary						antilog =	0.67	0.67
Model	R	R Square	Adjusted R^2		F	Sig.		
5	0.37	0.14	0.13		12.76	0.00		

However, level of education (bs = 1, some college = 0) and years of experience in software development affect the odds of finding a true answer for ease of use of software quality by a decrease of -0.33% and -0.06%. This implies that growing number of undergraduates (some college education = 0) and inexperience in software development (less years of experience) contribute to the decline of software quality or odds of not finding true ease of use of software quality. The constant suggests that ease of use software quality at beginning of the survey is 67.79%.

Using the average of each parameters in the observation, the probability (antilogs) of finding a true answer from BS respondents (= 1) and undergraduate (some college = 0) in favor of ease of use of software quality is 66.65% and 69.59%, respectively. This implies that respondents perceived that some college education affect the odds of keeping the ease of use of software quality. The component (factors) correlation matrix from the principal component analysis (PCA) (Table 4) shows that factors are associated at 55 – 68.2%. This correlation is true to all the other determinants of software quality from equation 1 – 5 in this study.

Table 4. Component Correlation Matrix from the PCA

Component	Correlation
Organizational	55.1
Technological	68.2
User-related	55

Maintainability versus Organizational, User-related and Technological Determinants

The maintainability equation can be expressed as:

Y_2 = maintainability, Ln (1/(1-P) = $b_0 + b_1x_1 + b_3x_3 + b_4x_4 + b_5x_5 + b_6x_6 + b_8x_8$

where:

x_1 = factor for organizational determinants (culture-resource-management)
x_3 = factor for technological determinants (services & reliability)
x_4 = age in years
x_5 = years of experience in software development
x_6 = gender: male = 1, female = 0
x_8 = level of education: bachelor of science (bs) = 1, some college = 0

Table 5 shows the determinants of maintainability of software quality. Although statistically insignificant, the constant indicates that the odds of finding a true answer for maintainability of software quality among the 405 observation is -0.005 (antilog = 49.88%). A 100% increase in organizational factors (company culture, time resource and process management), age in years, and technological factors (input-output services, type of database model used and network reliability) affects the odds in favor of finding true maintainability of software quality by 4.6%, 0.7%, and 3.0%, respectively.

Table 5. Logit Estimates for the Maintainability of Software Quality

Model	Parameters	Unstandardized Coefficients (B)	Standardized Coefficients (Beta)	t	Sig.	avg	bs = 1	some deg=0
6	(Constant)	0.00		-0.05	0.96	0.68	0.68	0.68
b1	culture resource mgt	0.05	0.22	4.24	0.00	8.26	0.38	0.38
b8	bs=1, some college = 0	-0.12	-0.20	-4.32	0.00	1.00	-0.12	0.00
b4	age in years	0.01	0.19	3.66	0.00	30.59	0.21	0.06
b3	service reliability	0.03	0.14	2.69	0.01	8.32	0.25	0.11
b5	years of experience in software dev	-0.02	-0.12	-2.45	0.01	3.47	-0.06	-0.49
b6	male=1, female=0	-0.05	-0.10	-2.20	0.03	1.00	-0.05	0.00
a	Dependent Variable: maintainability		0.13				1.28	0.73
Model Summary						antilog =	0.78	0.67
Model	R	R Square	Adjusted R^2	F	16.32			
6	0.44	0.20	0.19	Sig.	0.00			

However, the odds of finding true maintainability of software quality is affected negatively by level of education (some college = 0, -12.37%, years of experience in software development (-1.61%), and gender (female = 0, -5.45). This implies that the odds of finding true maintainability of software quality decline with undergraduates (some college = 0), younger software developer, and female respondents at 1% level of significance.

On the average, the odds of finding true maintainability of software quality is 78.29% in Bachelor of Science holders and 67.40% in undergraduates (some college = 0). The standardized beta coefficients indicate the exploratory power of the factors. In absolute values, the order of their of power shows that organizational factors (company culture, time resource and process management = 0.222), level of education (bs = 1, some college = 0 ; 0.199), age in years (= 0.186), technological factors (service & reliability = 0.140), years of experience in software development (= 0.121), and gender (male = 1, female = 0; 0.099) explain 96.8% of the odds of finding the true maintainability of software quality at 1% level of significance.

Reliability versus Organizational, User-related and Technological Determinants

The reliability estimates is expressed as:

Y_3 = reliability, Ln (1/(1-P) = $b_0 + b_1x_1 + b_3x_3 + b_4x_4 + b_5x_5 + b_8x_8$

where:

X_1 = factor for organizational determinants (culture-resource-management)
X_3 = factor for technological determinants (services & reliability)
X_4 = age in years
X_5 = years of experience in software development
X_8 = level of education: bachelor of science (bs) = 1, some college = 0

Table 6 exhibits the reliability of software quality. Holding the factors of software quality to zero, the constant suggests that odds of finding true (against false) reliability of software quality is 61.89% at the beginning of the survey. The odds of finding the true reliability of software quality increase by 1.25%, 1.05%, and 0.20% for every 100% increase in organizational factors (company culture, time resource and process management), technological factors (input-output services, type of database model used and network reliability) and age in years of respondents. This implies that the odds of finding true opinion of respondents on software quality are influenced by organizational and technological factors and mature individuals at 1% level of significance.

Table 6. Logit Estimates for the Reliability of Software Quality

Model	Parameters	Unstandardized Coefficients (B)	Standardized Coefficients (Beta)	t	Sig.	avg	bs = 1	some deg=0
5	(Constant)	0.48		14.88	0.00	0.48	0.48	0.48
b1	culture resource mgt	0.01	0.20	3.71	0.00	8.26	0.10	0.10
b8	bs=1, some college \0	-0.03	-0.14	-2.98	0.00	1.00	-0.03	1.00
b3	service reliability	0.01	0.16	2.99	0.00	8.32	0.09	0.09
b4	age in years	0.00	0.18	3.48	0.00	30.59	0.06	0.06
b5	years of experience in software dev	-0.01	-0.17	-3.41	0.00	3.47	-0.02	-0.02
a	Dependent Variable: reliability						0.69	1.71
Model Summary						antilog =	0.67	0.85
Model	R	R Square	Adjusted R^2	F	Sig.			
5	0.41	0.17	0.16	15.99	0.00			

The odds of finding true opinion of respondents on reliable software quality decline by -2.65% and -0.70% for every 100% of the opinion of undergraduates (some college = 0) with less years of experience in software development. This implies that odds of finding reliable software quality are low on college graduates (bs = 1) with less years of experience in software development at 1% level of significance. Using one (1) and zero (0) on the basis of level of education, average odds of finding true (against false) of reliable software quality are 66.5% and 84.71% for college (bs = 1) and undergraduate (some college = 0) individuals, respectively. This indicates that it is highly probable to find undergraduates with less experience in software development to answer true in determining reliability of software quality.

The coefficients beta explains the exploratory power of the factors of software quality. Organizational and technological factors explain 19.8% and 15.8% of finding the odds of reliability of software quality, respectively. Maturity of individuals (age in years) explains 18.0% of the odds of getting the true reliability of software quality. Level of education (some college =0) and years of experience in software development explain 2.7% and 0.7% of finding the probability of true reliability of software quality, respectively.

Relevance versus Organizational, User-related and Technological determinants

The relevance equation is expressed as:

Y_4 = relevance, $\text{Ln}(1/(1-P)) = b_0 + b_3x_3 + b_4x_4 + b_5x_5 + b_8x_8$

where:

X_3 = factor for technological determinants (services & reliability)

X_4 = age in years

X_5 = years of experience in software development

X_8 = level of education: bachelor of science (bs) = 1, some college = 0

Table 7 demonstrates relevance of software quality. If all factors are zero, the constant suggests that there is a probability of 70.25% on the odds of finding true (against false) of the relevance of software quality. The level of education (bs = 1, some college = 0) and years of experience in software development indicates that the odds of finding true relevance of software quality is 7.4% and 1.1%, respectively. This implies that individuals with Bachelor of Science degree (bs = 1) and more years of experience in software development tends to show favorable answer (true) on relevance of software quality unlike undergraduates (some college = 0). However, technological factors (input-output services, type of database model used and network reliability) and age of individuals decrease the odds of finding true relevance of software quality. This implies that technological factors and old age in finding the odds of relevance of software quality decline because of old technology and older individuals with no recent capability upgrading. With no improvement in technology and capability upgrading of aging individuals, software quality tends to be irrelevant.

On the average, there is an indication that the odds of finding relevance of software quality in individuals with college degree (bs = 1) is 66.71% against that of undergraduate individuals (65.06%). The findings also suggest that quality of software is relevant in individuals with more years of experience in software development.

Table 7. Logit Estimates for the Relevance of Software Quality

Model	Parameters	Unstandardized Coefficients (B)	Standardized Coefficients (Beta)	t	Sig.	Avg	bs = 1	some deg=0
4	(Constant)	0.86		14.51	0.00	0.86	0.86	0.86
b8	bs=1, some college = 0	0.07	0.21	4.27	0.00	1.00	0.07	0.00
b3	service reliability	-0.02	-0.20	-4.14	0.00	8.32	-0.20	-0.20
b5	years of experience in software dev	0.01	0.15	2.88	0.00	3.47	0.04	0.04
b4	age in years	0.00	-0.12	-2.20	0.03	30.59	-0.07	-0.07
a	Dependent Variable: relevance						0.70	0.62
Model Summary						antilog =	0.67	0.65
Model	R	R Square	Adjusted R^2	F	Sig.			
4	0.324236661	0.1051294	0.096180706	11.748002	4.934E-09			

Usefulness versus Organizational, User-related and Technological Determinants

The usefulness estimates is expressed as:

Y_5 = usefulness, $\text{Ln}\,(1/(1\text{-}P) = b_0 + b_1x_1 + b_3x_3 + b_4x_4 + b_5x_5 + b_8x_8$

where:

X_1 = factor for organizational determinants (culture-resource-management)

X_3 = factor for technological determinants (services & reliability)

X_4 = age in years

X_5 = years of experience in software development

X_8 = level of education: bachelor of science (bs) = 1, some college = 0

Table 8 shows the factors affecting usefulness of software quality. If all the factors are zero, the constant suggests that the odds ratio in favor of finding a “true” (against false) is 61.80% at the beginning of the survey. A 100% increase in organizational factors (company culture, time resource and process management) and technological factors (input-output services, type of database model used and network reliability) increase the odds ratio in favor of finding “true” by 1.3% and 1%. Maturity (age in years) of individuals increases the odds ratio of finding “true” by 0.2% at 1% level of significance. The findings imply that organizational and technological factors increase the probability of usefulness of software quality with mature individuals in the organization.

Table 8. Logit Estimates for the Usefulness of Software Quality

Model	Parameters	Unstandardized Coefficients (B)	Standardized Coefficients (Beta)	t	Sig.	avg	bs = 1	some deg=0
5	(Constant)	0.48		14.44	0.00	0.48	0.48	0.48
b1	culture resource mgt	0.01	0.20	3.76	0.00	8.26	0.11	0.00
b8	bs=1, some college = 0	-0.03	-0.14	-3.04	0.00	1.00	-0.03	-0.03
b3	service & reliability	0.01	0.15	2.93	0.00	8.32	0.09	0.09
b4	age in years	0.00	0.18	3.53	0.00	30.59	0.06	0.06
b5	years of experience in software dev	-0.01	-0.18	-3.53	0.00	3.47	-0.03	-0.03
a	Dependent Variable: usefulness						0.69	0.58
Model Summary						antilog =	0.66	0.64
Model	R	R Square	Adjusted R^2	F	Sig.			
5	0.41	0.17	0.16	16.27	0.00			

However, level of education (bs = 1, some college = 0) and years of experience in software development tend to decrease the odds ratio in favor of finding “true” in the observation. This implies that more undergraduates (some college = 0) with more years of experience in software development perceive that software quality is not useful (false) during the observation. Average odds ratio in favor of finding “true” is 66.49% if an individual is a college graduate (bs = 1) and 64.06% if an undergraduate (some college = 0). Organizational, technological factors and age explain 35.7% of the odds ratio of “true” in usefulness of software quality. All parameters (b0, b1, b3, b4, b5 and b8) are found to be significant at 1% level.

Summary of Results and Testing of Hypotheses

The results presented and can be statistically summarized below:

- The findings imply that Ease of Use of software quality is affected by organizational factors, technological factors and age in years of individuals. Level of education affects ease of use of software quality by 66.65% and 69.59%, respectively.

- Maintainability of software quality is affected by organizational factors (company culture, time resource and process management), age in years, and technological factors (input-output services, type of database model used and network reliability) at 4.6%, 0.7%, and 3.0%, respectively. The odd of finding true maintainability of software quality are 78.29% in Bachelor of Science holders and 67.40% in undergraduates.

- Reliability of software quality is influenced by the odds ratio in favor of finding true opinion of respondents on organizational and technological factors and mature individuals. Level of education indicates that findings true (against false) of reliable software quality are 66.5% and 84.71% for college and undergraduate individuals, respectively.

- Relevance is affected by the level of education and years of experience in software development indicating that the odds of finding true relevance of software quality are 7.4% and 1.1%, respectively. This implies that individuals with Bachelor of Science degrees and more years of experience in software development tend to show favorable answer (true) on relevance of software quality than undergraduates. There is an indication that the odd of finding relevance of software quality in individuals with college degree is 66.71%, against undergraduate individuals (65.06%). The findings also suggest that quality of software is relevant in individuals with more years of experience in software development.

- Usefulness is influenced by organizational factors, technological factors and age in years by 1.3%, 1% and 0.2%. The findings imply that organizational and technological factors increase the probability of usefulness of software quality with mature individuals in the organization. Average odds ratio in favor of finding "true" is "66.49%" if an individual is a college graduate and 64.06% if an undergraduate.

- Insignificant difference on the software development performance of undergrads as compared to bachelor of science degree holders of IT, computer science and information systems can be attributed to the following:

  1. **Difference in the curriculum of schools** – though the Commission on Higher Education (CHED) mandated certain subjects to be included in the curriculum of the schools offering IT, the fact that it is generic led to the slight difference. (E.g. if CHED mandated the schools to teach programming, it is still dependent on the schools what programming language they would be using considering the cost and all other resources involved).

  2. **Nature of the curriculum of IT in the universities** – Information Technology is a very broad field. Computer Science degree programs emphasize the mathematical and theoretical foundations of computing. A Computer Science degree is normally required in order to work in the Information Technology industry; even though it provides students a holistic learning in computer, most of the subjects are independent from each other. In order for a graduate to be a software developer, the main requirement would be knowledge and skills in programming, and that would be sufficient. Schools like AMA, STI, and Informatics are offering specialized courses that only deal with programming, thus, making them at par with the graduate of 4-year computer courses.

With the results from tests (Principal Component Analysis and Logistic Regression) together with cross validations on the related literatures and studies, the researcher concludes that the null hypotheses are all rejected.

CONCLUSIONS AND RECOMMENDATIONS

Conclusions

The following were the conclusions derived that would answer the research questions of this study:

1. *What is the influence of organizational factors on software quality of information system projects?*
   The quantitative findings, together with reviewed literatures both implied that software quality is influenced or affected by the organizational factors or determinants of software quality which are Company Culture, Time Resource, and Process Management.

2. *What is the influence of technological factors on software quality of Information System projects?*
   The quantitative findings, together with reviewed literatures both implied that software quality is influenced or affected by the technological factors or determinants of software quality which are Input-Output Services, Type of Database Model Used, and Network Reliability.

3. *What is the influence of user-related factors on software quality of Information Systems projects?*
   The quantitative findings, together with reviewed literatures both implied that software quality is influenced or affected by the user-related factors or determinants of software quality which are Level of User Involvement, User Resistance to Change, and User's training on Systems.

4. *What is the strength of association between the determinants of software quality and Ease of Use as a software quality attribute?*
   The findings implied that ease of use of software quality is affected by organizational factors (Company Culture, Time Resource and Process Management) and technological factors (Input-Output Services, Type of Database Model Used and Network Reliability), together with age in years of individuals and level of education. Moreover, the results showed that years of experience and level of education are directly proportional to software quality. Software Developers with more years of experience and higher level of education tend to develop software with an ease of use characteristic; and ease of use significantly decreases as the years of experience and level of education decrease. Therefore, the researcher concludes that Organization and Technological determinants, together with Level of Education and Software Development Years of Experience are positively correlated with each other in terms of Ease-of-Use. The finding not only implied positive correlation but also a strong positive correlation.

5. *What is the strength of association between the determinants of software quality and Maintainability as a software quality attribute?*
   The findings implied that maintainability of software quality is affected by organizational factors (company culture, time resource and process management), together with age in years; and technological factors (Input-output services, type of database model used and network reliability) with input and output services as the greatest factor that has influence. However, true maintainability is negatively affected by level of education, experience in software development and gender (female). Therefore, the researcher concludes that in finding true maintainability, organizational factors and technological factors must be considered, together with age in years of developers, level of education, years of experience and gender; as they are all positively correlated with each other.

6. *What is the strength of association between the determinants of software quality and Reliability as a software quality attribute?*
   The findings implied that in terms of reliability, there is a constant increase in software quality as the organizational (company culture, time resource and process management), together with age in years and technological factors (Input-output services, type of database model used and network reliability) also increases. The odds of finding reliability of software quality decline with undergraduates (less education), and with those who have less years of experience in software development. Therefore, the researcher concluded that software quality is being influenced by organizational and technological

factors, maturity of individuals, level of education, and years of experience in software development. All these variables are positively correlated with each other.

7. *What is the strength of association between the determinants of software quality and Relevance as a as software quality attribute?*
   Based on the findings, the researcher concludes that in terms of relevance, there is a high probability on the odds of finding true relevance of software quality with developers having higher level of education (Bachelor of Science degree holders) and with more years of experience in software quality as compared to undergraduates and with less years of experience in software development. Technological factors (input-output services, type of database model used and network reliability) and age of individuals are negatively correlated to each other. These implied that older individuals have less impact on the relevance of software as compared to younger individuals. Aging individuals, those with no improvement in technology and capability upgrading, software quality tends to be irrelevant.

8. *What is the strength of association between the determinants of software quality and Usefulness as a as software quality attribute?*
   Based on the results of this study, the researcher concludes that organizational factors (Company Culture, Time Resource and Process Management) and technological factors (Input-Output Services, Type of Database Model Used and Network Reliability), increase the probability of usefulness of software quality with matured individuals in the organization. However, level of education and years of experience in software development tend to decrease the odds ratio in favor of finding "true" in the observation. These imply that more undergraduates with less years of experience in software development will negatively provide usefulness attribute of software quality. All parameters (b0, b1, b3, b4, b5 and b8) are found to be significant at 1% level.

Recommendations

As supported by the findings and conclusion, the following are the plausible recommendations towards achieving the main motivation of the study:

1. This study, with the aid of other related literature, made firm the importance of Software Quality in organizations. Organizational factors (Company Culture, Time Resource and Process Management), Technological factors (Input-Output Services, Type of Database Model Used and Network Reliability), User related factors (Level of User Involvement, User Resistance to Change and User's training on Systems) all have influence on software quality; thus, the researcher recommends that organizations which are into software development should put focus on the development and inculcation of these determinants and these determinants should not be neglected as the findings are statistically significant.

2. This study made firm the importance of education in software quality. Results showed that developers with Bachelor of Science degrees tend to provide more quality on the software that are being developed. The researcher therefore recommends that in terms of hiring future software developers, HR should prioritize the hiring of Bachelor of Science degree holders, if quality is the focus.

3. This study made firm the importance of software development experience in software quality. Results showed that developers with more development experience tend to provide more quality on the software that are being developed. The researcher thus recommends that in terms of hiring future software developers, HR should prioritize the hiring of experienced developers, if quality is the focus.

4. This study, together with the literature reviewed showed the importance of software quality in organizations. Poor strategic alignment between business and IT strategies can result in a failure of the organization to reach its full potential. The researcher therefore recommends the alignment of IT strategy and business strategy. Because aligning business strategy with IT strategy tends to have a positive impact on organizations, strategic alignment should be a top priority for senior managers. They should regard IT as a way of meeting business goals and providing value.

In order to further enrich the knowledge obtained from this study, the proponent suggests the following:

1. To increase the data that will be gathered from the respondents in terms of demographics, the type of programming languages that they are currently using and/or have experience in using, the type of databases that they are using, and the nature of the company that they are working for. Through this, the researcher would be able to further correlate the software quality attributes to the demographics of the respondents, thus, more data can be derived.

2. For the future researchers, to re-evaluate the variables to be used, the points to be considered, and the journals to be referenced because technology is consistently changing and advancing.

3. For future researchers, to consider the current status of the software industry in the Philippines (if it be the country in focus). Forecast coming from credible organizations such as the Philippine Software Industry Association said that IT industry is booming in the Philippines and is expected to grow more in the coming years; thus, future studies on this topic should be up to date.

4. To consider using ISO/IEC 25010:2011 (Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models) as a basis for software quality attributes/variables in the study.

STRATEGIC IMPLICATIONS

The following strategic policy implications were identified.

- **Industry and Finance:** The results obtained totally agreed with what Bundschuh and Dekkers (2008), stated on their study. According to them, in today's brutally tough software marketplace, it is of great importance to quantify software quality and process output. Jones and Bonsignour (2012) said in their study on the Economics of Software Quality, that a high level of software quality will raise the economic value of software for the producers, financiers, and the consumers of the software applications. Poor quality software costs more to build and to maintain than high-quality and it can also degrade operational performance, increase user error rates, and reduce revenue by decreasing the ability of employees to handle customer transactions or attract additional clients. In the software industry, not only is quality free, but also benefits the entire economic situations of both developers and clients.

  Additionally, results of this study affirmed what the PSIA (2010), is saying that there is a big opportunity for the Philippines in terms of software development. This study also made firm the conclusion of the study by Parcia (2011), that software industry in the Philippines has not reached its full potential. The young ages of the respondents are evidence of the industry’s infancy. System quality assurance competencies are considered barometers in the determination of the types of projects that Philippines software developers can handle. Thus, it could be suggested that focus on software quality in the Philippine setting could be of great help.

- **Academic Community:** In the Philippines, there is an abundance of degrees that point to the role of software developers. With the obtained results from this study, it can be seen that it is not always the technological factors that matter. The universities must produce graduates who excel not only in terms of technical knowledge and know-how but also in terms of sophistication, adaptability, and professionalism. The curriculum should take students beyond the traditional confines of computer science and information technology by exposing them to an environment oriented to both business and technology encouraging the students to be more culturally aware and to develop values to be open for change and to be driven without compromise.

REFERENCES

Aktas, E., Cicek, I., & Kiyak, M. (2011). The effect of organizational culture on organizational efficiency: The moderating role of organizational environment and CEO values. *Procedia Social and Behavioral Sciences, 24*, 1560-1573.

Ambler, S. (2013). *The "Broken Iron Triangle" Software Development Anti-pattern, Ambysoft: Effective Practices for Software Solution Delivery*. Retrieved September 5, 2013, from http://www.ambysoft.com/essays/brokenTriangle.html

Ambrose, P. J., & Chiravuri, A. (2010). A socio-cognitive interpretation of the potential effects of downsizing on software quality performance. *Information Systems Journal*, *20*(3), 239-265.

Barney, S., Mohankumar, V., Chatzipetrou, P., Aurum, A., & Wohlina, C. (2013). Software quality across borders: Three case studies on company internal alignment. *Information and Software Technology*, *56*(1), 20-38.

Bundschuh, M., & Dekkers, C. (2008). *The IT measurement compendium estimating and benchmarking success with functional size measurement*. Berlin: Springer.

Bush, W. R. (2007). Software, regulation, and domain specifity. *Information and Software Technology*, *49*(1), 44-54.

Cimperman, R. (2006). *UAT defined a guide to practical user acceptance testing as a silver bullet*. Upper Saddle River, N.J.: Addison Wesley Professional.

Davidson, J. O. C. (1994). The sources and limits of resistance in a privatized utility. In K. D. Jermier and W. R. Nord (Eds.), *Resistance and power in organizations* (pp. 69–101). London: Routledge.

Ezamly, A., & Hussin, B. (2011). Estimating quality-affecting risks in software projects. *International Management Review*, *7*(2), 66-74.

Gorla, N., & Lin, S. (2010). Determinants of software quality: A survey of information systems project managers. *Information and Software Technology*, *52*(6), 602-610.

Gujarati, D. (1999). *Essentials of econometrics* (*2nd edition*). Boston, USA: McGraw-Hill International Editions.

Hambling, B., & Goethem, P. (2013). *User acceptance testing: A step-by-step guide*. Swindon: BCS Learning & Development Limited.

Hanssen, G. K. (2012). A longitudinal case study of an emerging software ecosystem: Implication for practice and theory. *The Journal of Systems and Software*, *85*(7), 1455-1466.

Hauge, O., Ayala, C., & Conradi, R. (2010). Adoption of open source software in software-intensive organizations: A systematic literature review. *Information and Software Technology, 52*(11), 1133-1154.

Hazzan, O., & Hadar, I. (2008). Why and how can human-related measures support software development processes. *The Journal of Systems and Software*, *81*(7), 1248-1252.

Jones, C., & Bonsignour, O. (2012). *The economics of software quality*. Upper Saddle River, NJ: Addison-Wesley.

Klaus, T., & Blanton, J. E. (2010).User resistance determinants and the psychological contract in enterprise system implementations. *European Journal of Information Systems*, *19*(6), 625-636.

Kujala, S. (2003). User involvement: A review of the benefits and challenges. *Behaviour & Information Technology*, *22*(1), 1-16.

Li, Y., Chang, K. C., Chen, H., & Jiang, J. (2010). Software development team flexibility antecedents. *The Journal of Systems and Software*, *83*(10), 1726-1734.

McKendrick, J. (2013). *6 shining examples of cloud computing in action - Forbes*. Retrieved September 10, 2013, from http://www.forbes.com/sites/joemckendrick/2012/02/22/6-shining-examples-of-cloud-computing-in-action/

Nagadevara, V. (2008). Methodology for identifying factors that impacts software quality. *Journal of Academy of Business and Economics*, *8*(2), 145-150.

Parcia, R. (2011). Software engineering and development in the Philippines: The competencies of software developers and engineers. *University of Santo Tomas - Graduate School*, 47-51.

Philippine Software Industry Association (PSIA). (2010). *How to join*. Retrieved August 23, 2012, from http://psia.org.ph/MEMBERSHIP/HowtoJoin.aspx

Princeton University. (2013). *Garbage in, garbage out - Princeton University - Home*. Retrieved September 7, 2013, from http://www.princeton.edu/~achaney/tmve/

Rob, P., & Coronel, C. (2009). *Database systems: design, implementation, and management* (*8th edition*). Boston, MA: Cengage Learning.

Rothenberger, M., Kao, Y., & Wassenhove, L. V. (2010). Total quality in software development: An empirical study of quality drivers and benefits in Indian software projects. *Information and Management, 47*(7-8), 372-379.

Sarngadharan, M. (2010). *Management information system* (*revised edition*). Mumbai, India: Himalaya Publishing House.
Scarpa, M., & Puliafito, A. (2009). Developing high quality software. *International Journal of Parallel, Emergent and Distributed Systems*, *24*(2), 171-187.
Shoemaker, C. (2009). *Secrets of real world ASP.NET dynamic data websites*. Indianapolis, IN: Wiley.
Yaldex.com. (2013). *The importance of good database design*. Retrieved September 6, 2013, from http://www.yaldex.com/php_tutorial_3/ch15lev1sec1.html
Trkman, P. (2010). The critical success factors of business process management. *International Journal of Information Management*, *30*(2), 125-134.
Vavpotic, D., & Bajec, M. (2009). An approach for concurrent evaluation of technical and social aspects of software development methodologies. *Information and Software Technology*, *51*(2), 528-545.
Wang, H., Khoshgoftaar, T., Hulse, J. V., & Gao, K. (2011). Metric selection for software defect prediction. *International Journal of Software Engineering and Knowledge Engineering*, *21*(2), 237-257.
Wilderom, C., Den Berg, P. V., & Wiersma, U. (2012). A longitudinal study of the effects of charismatic leadership and organizational culture on objective and perceived corporate performance. *The Leadership Quarterly*, *23*(5), 835-848.
Wong, E. Y., & Tate, G. (1994). A study of user participation in information systems development. *Journal of Information Technology*, *9*(1), 51-60.
Zehir, C., Ertosun, G., Zehir, S., & Müceldili, B. (2011). The effects of leadership styles and organizational culture over firm performance: Multi-national companies in Istanbul. *Procedia Social and Behavioral Sciences*, *24*, 1460-1474.

Author's Biography

Jerhmie James A. Zacarias is an experienced Project and Technology Management Professional. He is currently working as a Technology Relationship Manager for Wells Fargo, the second-largest bank in the world in terms of market capitalization, where he manages IT Projects and Service Delivery. Other experiences include being a Regional Lead (Europe) for IT Client Service Management, Project Manager and Business Analyst within the Banking, Business Process Outsourcing, Engineering, and IT consulting industries. He holds a Bachelor of Science in Information Technology degree from the University of the East and Master of Science, cum laude and bene meritus, in Management Engineering degree from the University of Santo Tomas. He was a recipient of an international scholarship grant from the Project Management Institute Educational Foundation, Houston, USA Chapter.