Factors Affecting Product Quality in the Software Development Industry of Sri Lanka

Sahan Jayawarna
A.T. Fonseka

Abstract

The software development industry of Sri Lanka has grown rapidly during recent years. However, the quality of software products has become a critical issue as shown in the increase in software defects and associated solutions for those problems. This study aimed at understanding the critical factors underlying software product quality. It was based on a survey of ten software development companies and two software development units in non-software development companies. The sample consisted of 150 respondents. The study showed that it is necessary to have (i) technological excellence (ii) software process maturity (iii) team productivity and (iv) stakeholder involvement in order to deliver high quality products. Failure to deliver high quality software products will result in higher software life cycle costs and loss of future sales revenue.

Key Words : *Sri Lanka's Software Development Industry, Software Product Quality*

1. Introduction

The global IT industry has been expanding rapidly over the past two decades. The US financial and economic crisis of 2008 slowed down the progress of the industry but it now shows signs of recovery. Software solutions are used in almost all industries ranging from the retail trade to automotive, chemicals, energy, utilities, to aerospace. Hence, it is an industry with great potential.

The Sri Lankan software industry, even though small, has grown impressively during the past fifteen years. At present, there are over 175 software development companies

Sahan Jayawarna, B.Sc. (Computer), CIMA (UK), BCS (UK,) MBCS, and an MBA from PIM, is a software engineer at Dialog Telekom Ltd.

Prof. A. T. Fonseka, is a senior faculty member at the Postgraduate Institute of Management (PIM), Colombo, Sri Lanka.

in Sri Lanka actively involved in the development of software products, and providing services for both export and domestic markets. Export earnings from software and IT-enabled services which stood at US$ 5.0 million in 1996 rose to US$ 275 million in 2009. In 2005, the Software Exporters' Association (SEA) targeted to achieve US$ 1 billion from software development by 2012. Due to the slow recovery in the global economy this date is likely to be deferred to 2016 according to reliable industry sources.

Despite the progress achieved, a major problem afflicting the industry relates to software product quality. Software quality is defined as the totality of functions and features of a software product that bears on its ability to satisfy stated or implied needs. This study purports to address issues relating to software product quality.

This article begins with a description of the research problem and a review of the literature relevant to the problem. This is followed by the conceptual framework and the study design. The data gathered during the field study are presented next, followed by the study findings and discussion of the findings. The final section contains the conclusions of the study and recommendations.

2. Research Problem

The Sri Lankan software industry has shown impressive growth since 1996, but the industry is under extreme pressure to meet the requirements of demanding customers. Although the importance of software quality has increased, the level of software quality has not improved. Software companies meanwhile are working hard to produce better quality software. The costs of poor software quality in terms of reduced productivity, downtime, and customer dissatisfaction can be enormous.

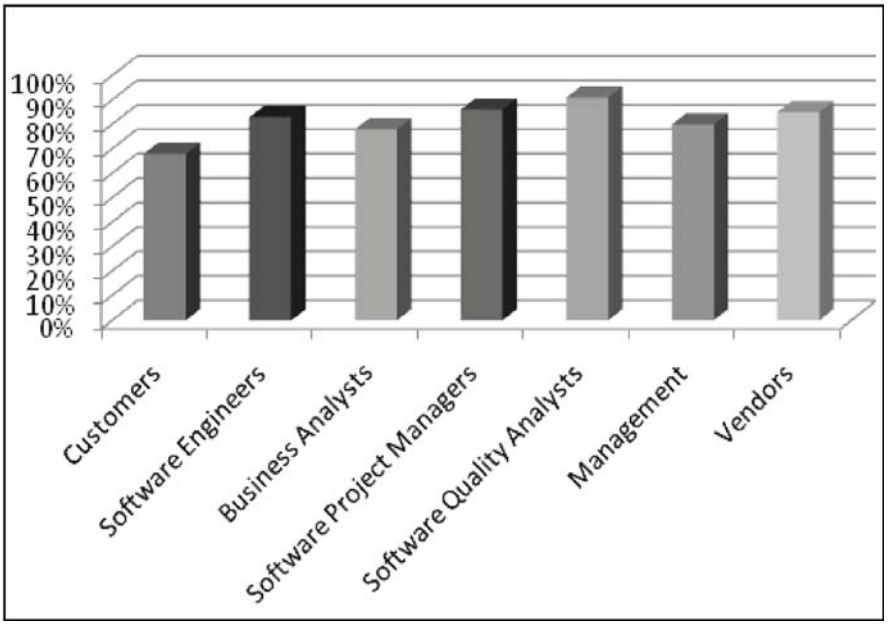
The software crisis is a situation that has existed since the late 1970s, characterized by the inability of software developers to deliver high quality software on time and on budget. At a CIO forum in New York during the early part of the decade, more than 40% of the CIOs in attendance admitted that their companies had not paid adequate attention to the quality of their software (A software quality crisis is brewing, 2008). The senior-level executives who participated in the CIO forum belonged to corporations that had revenues in excess of $ 5 billion and IT budgets approaching a billion dollars. More than 40% of all software applications had been released with critical defects ranging from one to ten, with management being fully aware of this at the time of issue. Colin Armitage, the CEO and the founder of the Original Software Group, observed that "with software increasingly underpinning business-critical applications, sloppy attitudes toward quality from the people who make it are no longer acceptable."

Since software is now used in many demanding applications, poor software quality has become an issue of great concern as it has many negative consequences. Poor software product quality is mainly seen in (i) customer dissatisfaction, (ii) increase in service packs and patches, (iii) system downtime in information systems, and (iv) difficulty in running on different operating systems/platforms. The purpose of this

study was to identify the factors that contribute to the poor quality of software products in the Sri Lankan software industry.

A pilot survey was conducted among fifty respondents to identify the level of quality of software products as perceived by different stakeholders. Customers, software engineers, business analysts, software project managers, software quality analysts, senior managers and vendors of external suppliers were included in the pilot survey. The results are shown in Figure 1.

Fig 1: Level of Quality in Software Products



All of the software development teams that participated in the pilot survey had either a separate quality assurance team or quality analysts. Ninety-five percent of the respondents admitted that it is impossible to deliver a totally bug-free software product.

3. Objectives

The objectives of the study are:

A. To identify the key factors contributing to poor software quality in the software industry of Sri Lanka,

B. To make recommendations for improving software quality.

4. Significance of the Study

This study is significant from two angles, i.e., contribution to the body of knowledge and its practical importance for management. To the best of the researcher's knowledge, no research has been done previously on this topic in the Sri Lankan context. Thus, the study will add to the research literature.

Further, the study will be of practical importance for management. Software products are deep-rooted in almost every aspect of our daily lives and most businesses would find themselves unable to trade without them. The situation has grown to such an extent that a small glitch in a software application or a computer outage can cost companies vast sums of money and in some cases even put them out of business. Thus, identification of the factors which affect software quality and the presentation of implementable recommendations will have great practical value. Additionally, companies can focus on better strategies for the management of quality and improvement of software quality which will lead to customer satisfaction as well as a saving of time and cost.

5. Literature Review

Technological Excellence

There are two main categories of software technologies and tools, i.e., open source and closed source software products. Open Source Software (OSS) is defined as computer software for which the source code and certain other rights normally reserved for copyright holders are provided under a software license that meets the open source definition or that is in the public domain (Open Source, 2009). Some of the open source software includes Linux, GNOME, Apache, Firefox, Dovecot, Open Office, Moodle and MySQL. 'Closed source' is a term for software whose license does not allow for the release or distribution of the software's source code. Generally, it means only the binaries of a computer programme are distributed and the license provides no access to the programme's source code (Closed source software, 2009). Well-known examples of proprietary software include Microsoft Windows, Macromedia Flash, Macromedia Dreamweaver, Adobe Photoshop, Mac OS, WinZip, Oracle, etc. Thus, software products can be developed either using open or closed source technologies, tools, or both.

In a study by Abreu and Melo (1996), the impact of Object-Oriented design on software quality characteristics was experimentally evaluated. They adopted a suite of metrics for Object-Oriented design to measure the use of Object-Oriented design mechanisms, and data were collected on the development of eight small-sized information management systems. Object-oriented design is the process of planning a system of interacting objects for the purpose of solving a software problem. An object contains encapsulated data and procedures grouped together to represent an entity. The five basic concepts of object-oriented design are the implementation level features that are built into the programming language, i.e., class, information hiding, inheritance,

interface and polymorphism. Abreu and Melo (1996) showed that Object-Oriented design mechanisms such as inheritance, polymorphism, information hiding and coupling can influence quality characteristics like reliability or maintainability.

The results of the study done by Low and Huan (1999) suggest that the quality of software developed using CASE (Computer-Aided Software Engineering) tools is better than conventionally developed systems with respect to reliability, maintainability and portability, but the degree of improvement is affected by the particular CASE tool used for development. CASE refers to software used for the automated development of systems software. Some of the typical CASE tools are configuration management tools, data modeling tools, model transformation tools, program transformation tools, refactoring tools, and source code generation tools.

Process Maturity

There are four main processes in software development, i.e., business analysis, software engineering, quality assurance and project management. Business analysis involves identifying business needs and determining software solutions to business problems. This includes requirement identification, communication, assessment and validation. Software engineering is the application of engineering to software and development of software products. Software quality assurance involves monitoring the software engineering processes and products used to ensure quality. Software project management includes planning, leading and controlling software projects since software development is usually carried out as projects.

Quality in software design predicts quality in later stages of the product life cycle. Fenton and Neil (1999) stated that quality must be designed or built into software products from the start. There is some empirical evidence to support this notion from Japanese software factories, the NASA Space Shuttle programme, and a variety of projects at the IBM (Harter and Slaughter, 2000: 407). Software design is a process of problem-solving and planning for a software solution. After the purpose and specifications of software are determined, software developers will design or employ designers to develop a plan for a solution. It includes low-level component and algorithm implementation issues as well as the architectural view. Software designers or architects may identify a design problem that has been solved by others before. Those design patterns describe solutions to common problems. The re-use of such patterns can speed up the software development process as they have been tested and proved in the past.

There is some empirical support linking process maturity to software quality. For example, Diaz and Sligo (1997) found initial evidence of a positive relationship between process maturity and software quality at Motorola. Fenton and Neil (1999) stated that when a process becomes more mature and less variable, the outputs of the process (i.e., software products) have a higher level of quality. This implies that the maturity of the software development process is essential to reduce process variability and thus improves the quality of software products. Determining whether process maturity is

linked to software quality is important because, in practice, many managers still emphasize testing at the end of the development cycle instead of building in quality through better processes.

Harter and Slaughter (2000) developed a conceptual framework for assessing the relationship between process maturity and software quality at different stages of the product life cycle, i.e., development, implementation and production. They identified the direct and indirect marginal effects of improved process maturity on software quality at different stages of the software life cycle. Quality at each stage increases as a direct benefit of higher maturity, but also improves due to quality built into the product at earlier stages. This means a higher level of process maturity leads to higher software quality. Software development is carried out through project teams. Boehm and Kemerer (1997) argued that the quality of the software team (i.e., the capability of programmers and analysts) is the second most significant factor in determining either cost or quality of software products.

Team Productivity

A study done of leading European software organizations identified that the quality of people in software teams is one of the most important factors in improving productivity and quality of software projects (Blackburn et al., 1996). A study by Mazhil Rajendran (2005) analyzed team effectiveness in software development teams working on hardware and software environments using the Belbin self-perception inventory. The Belbin team role inventory assesses how an individual behaves in a team environment. The research study presents a team effectiveness analysis of software development teams. The researcher found that Belbin's roles and the Belbin self-perception inventory can be used to recognize important positive and negative features of a software development team, and thus encourage positive features and avoid ad remedy negative features.. Further, it was concluded that forming teams based on *who can work effectively together* will improve the effectiveness of software development teams.

Another study Hye-Ryun Kang, Hee-Dong Yang and Chris Rowley (2006) investigated the importance of team member characteristics, particularly cognitive and demographic, on team effectiveness in software development. Their analysis showed that team effectiveness is influenced more by cognitive rather than demographic similarities.

Krishnan (1998) identified that higher levels of domain and language experience in the team result in more familiarity with the application domain, and the syntax and semantics of the programming language used, and thus lead to higher software quality. The study indicated that personnel capability of the team is significantly associated with a lower number of defects and enhanced productivity in packaged software products.

A study by Acuña, et al. (2008), examined the work climate within software development teams and whether the team climate in software development teams has any relation

to software product quality. They defined team climate as the shared perceptions of the team's work procedures and practices. The team climate factors examined were participative safety, support for innovation, team vision and task orientation. They found that high team vision preferences and high participative safety perceptions of the team were significantly related to better quality software.

The founder of the Software Process Program of the Software Engineering Institute Watts S. Humphrey says: "to get a quality product you need two things: programmers personally committed to producing high-quality stuff, and managers and customers that demand high-quality work". Active customer involvement is crucial to any software development project. Customer involvement should be from concept to delivery of the project. It helps them to get better visibility of the development process and its problems, and a better idea of the progress being made. There are mainly two types of customers in a software project, i.e., individuals and organizations. When the customer is another organization, software products are more complex and large. However, the customers in agile software projects are involved to a greater degree than in traditional projects.

Stakeholder Involvement

More customer involvement promotes less chance of misunderstanding of requirements and promotes building the right working software for customers according to their expectations, which in turn helps to get more business from customers. It is necessary to consider the customer as a part of the software development team who is equally responsible for project success or failure, by building trustworthy relationships and providing right visibility at the right time. As Burns and Halliburton (1989) identified, when coupled with strong customer involvement and living requirements, the system evolves to cleaner, simpler designs that approach the original customer vision.

External vendors play a vital role in developing the software products according to the desired quality. For example, most of the hardware such as servers for software projects is bought from external vendors. If these hardware systems are not compatible with the software developed due to the lack of engagement between vendors and software development team, then the resultant software product will be of poor quality.

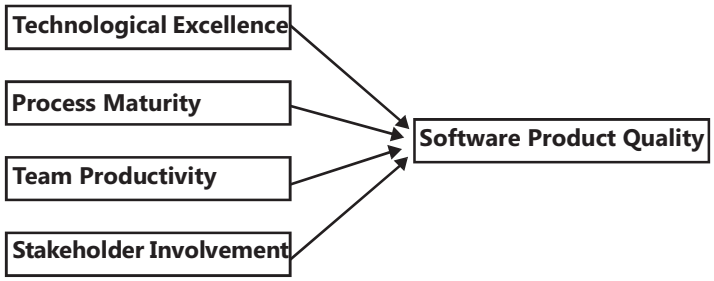
Zwikael (2008) conducted a study on top management support for projects executed in the software sector. The research was carried out in Japan, Israel and New Zealand. The objective of this study was to identify those top management support processes that have the greatest impact on software development project success, and to compare these critical processes with the actual type of support provided by organizations. It has been found that in all three countries top managers do not invest adequate effort in critical processes, compared to non-critical ones. Further, critical top management support processes that have a higher impact on project success, often do not receive an appropriate level of attention from senior managers in the software industry. It was recommended that senior managers in the software industry should focus on critical processes rather than on less important ones.

According to the literature review, the criteria for software product quality are (i) technological excellence, (ii) software process maturity, (iii) team productivity, and (iv) stakeholder involvement. The literature review provides the logical foundation for the conceptual framework that follows.

6. Conceptual Framework

The foregoing review of literature showed that four key factors affect software quality. These are illustrated in the conceptual framework below (Figure 2).

Fig 2: Conceptual Model



Source: Conceptualization by the Researcher

7. Hypotheses

The following hypotheses are formulated based on the conceptual model:

H1: The higher the technological excellence, the higher the software product quality.

H2: The higher the maturity of the software process, the higher the software product quality.

H3: The higher the productivity of software development teams, the higher the software product quality.

H4: The higher the stakeholder involvement in software development projects, the higher the software product quality.

8. Study Design

The concepts contained in the hypotheses were reduced to their operational terms, i.e., to their variables, indicators, and measures. Table 1 shows the process of conceptual reduction.

Table 1: Process of Conceptual Reduction

Concept	Variable	Indicator	Measure
Technological excellence		Design methodology and tools	Presence of design tools, Attitude towards the use of software designs
		Development tools selection and performance	Presence of built-in compilers, Use of integrated software development frameworks
	Hardware	Infrastructure stability	Availability of backup systems, disaster recovery sites
		Capacity	Memory, Storage capacity, Processing speed
		Inter-operability	Work with a wide range of other systems, Networking capability
Process maturity	Business analysis	Requirement gathering extent	Requirement gathering meetings with clients
		Analysis tool usage	Applicability of the analysis tool
		Documentation practices	Functional specifications should contain most user requirements
	Programming	Application of programming practices	Appropriate selection of programming models, Whether code reviews are done and frequency, Appropriate naming conventions, Modularity of systems, Willingness to follow best programming practices
		Documentation practices	Completion of technical specifications
	Quality assurance	Coverage of testing	Test cycles and length, Test types used, Presence of smoke testing, Availability of test cases
		Application of test automation	Types of test automation tools used, Use of testing tools
		Documentation practices	Issuance of release note with no of test cases passed, on-hold, failed, Areas which are not tested due to other system dependencies

	Project management	Project score	Completed, delayed on-hold projects
		Work allocation	Employees are neither idle nor overloaded
		Documentation practices	Presence of a project plan, identification of critical path and near critical path activities
Team productivity	Efficiency	Time	Whether projects are completed on-time
		Cost	Whether projects have exceeded the budgets
	Effectiveness	Composition	Different roles and diversity of the team
		Context	Availability of adequate resources, Leadership style
		Work design	Presence of task autonomy, skill variety, task significance
Stakeholder involvement	Customer	Customer interaction	No. of review meetings with customer, No. of steering committee meetings with customer
	Management	Management support	Availability of objectives for software projects, Setting adequate budgets and time plans, Attitude of management
	Vendors	Vendor engagement	No. of meetings with vendor, Site visits by vendor
Software product quality	Reliability	Defects	No. of defects/software bugs
		Transaction volume	Response time
	Usability	Application security	Network and application security level
	Maintainability	System modularity	Ability to incorporate change requests
	Portability	Device independence	Performance in different platforms
	Availability	System up time	Time period in which system runs without any failures

Source: By the Researcher

The main instrument for data collection was a questionnaire consisting of 5-point Likert scale statements ranging from strongly agree to strongly disagree. Statements were formulated for all variables considered in the study and randomly ordered so that variables do not appear sequentially. Then, background information of the respondents was sought in the questionnaire. A very unfavourable response to a question was given 1 point, while a very favourable response was given 5 points. The questionnaire was hosted online making it convenient for the respondents to answer. The research was conducted in the Western Province. Data was collected from software development companies as well as from in-house software development units within non-software development companies. For example, data was collected from software development companies such as the IFS, MillenniumIT, Virtusa, etc. as well as from in-house software development units in Dialog and the Nations Trust Bank. The questionnaire was supplemented by interviews containing open-ended questions. This helped to ensure triangulation.

9. Data Presentation and Analysis

The population consisted of software development stakeholders in the Western Province. Ten software development companies and two software development units of non-software development companies were chosen. Questionnaires were sent to the software professionals. Since the questionnaire was hosted online, the URL was sent to the respondents most of the time. The survey was closed when 150 responses were received.

Table 3 shows the educational background of the respondents. Seventy four (74) respondents out of hundred and fifty (150) possessed basic degrees while forty six (46) had postgraduate qualifications. Further, seventeen (17) respondents had professional qualifications.

Table 3: Education level Analysis of Sample

Education level	Frequency	Percentage
Graduate	74	49%
Postgraduate	46	31%
Professional	17	11%
Secondary school	3	2%
Undergraduate	10	7%
Total	150	100%

Source: Survey data

Table 4 gives a job-wise analysis of the respondents. Most of the respondents in the sample were executives, and accounted for 66% of the sample. 20% of the sample represents the respondents who are at managerial level. Since a flat organization hierarchy is required for software development, this composition is justified.

Table 4: Job-wise Analysis of Sample

Job category	Frequency	Percentage
Executive	99	66%
Managerial	30	20%
Administrative	1	0.7%
Other	20	13.3%

Source: Survey data

The following Table shows the cross tabulation of data between education level and gender.

Table 5: Analysis of Sample by Education level and Gender

		Education level					Total
		Graduate	Postgraduate	Professional	Secondary school	Undergraduate	
Gender	Female	23	13	3	0	3	42
	Male	51	33	14	3	7	108
Total		74	46	17	3	10	150

Source: Survey data

There were more male respondents than female respondents at each educational level in the sample.

Table 6 presents the cross tabulation of data between job and gender.

Table 6: Analysis of Sample by Job category and Gender

		Job category				Total
		Administrative	Executive	Managerial	Other	
Gender	Female	0	35	2	5	42
	Male	1	64	28	15	108
Total		1	99	30	20	150

Source: Survey data

It clearly shows that there are more male respondents than female respondents in each job category.

Validity

Validity ensures that the measuring instrument (the questionnaire) measures the concept that is intended to be measured. The questionnaire was tested for convergent validity. Factor analysis was done using the principal component analysis extraction method for all the variables separately. It has resulted in extracting one component which indicates the presence of convergent validity.

Reliability

Reliability of a measure indicates the extent of absence of bias (error-free). It is an indication of the stability and consistency with which the instrument measures the concept and helps to assess the "goodness" of a measure. Hence, the responses were checked for reliability, and measured using Cronbach's Alpha. It is the coefficient of reliability and a minimum value of 0.7 was accepted. Table 7 shows Cronbach's Alpha values for each variable in the study.

Table 7: Reliability of the Sample

Variable	Cronbach's Alpha	Number of statements
Technological Excellence	0.765	6
Process Maturity	0.835	10
Team Productivity	0.781	15
Stakeholder Involvement	0.642	6
Software Product Quality	0.794	8

Source: Survey data

As presented in Table 7, all variables have Cronbach's Alpha values greater than 0.7. Therefore, it can be concluded that there is a high level of reliability in the sample.

Testing of hypotheses

H1: The higher the technological excellence, the higher the software product quality. It is accepted because the correlation analysis showed that there is a positive correlation between technological excellence and software product quality (+ 0.432 at 0.01 significance level).

H2: The higher the maturity of the software process, the higher the software product quality. It is accepted because the correlation analysis showed that there is a positive correlation between software process maturity and software product quality (+ 0.600 at 0.01 significance level).

H3 : The higher the productivity of software development teams, the higher the software product quality. It is accepted because the correlation analysis showed that there is a positive correlation between team productivity and software product quality (+ 0.553 at 0.01 significance level).

H4: The higher the stakeholder involvement in software development projects, the higher the software product quality. It is accepted because the correlation analysis showed that there is a positive correlation between stakeholder involvement and software product quality (+ 0.617 at 0.01 significance level).

10. Findings

The findings from the ten interviews are summarized below:

All respondents stated that they were not satisfied with regard to software product quality, either to some extent or totally. Most of the managers in software development teams also stated that they were not fully satisfied with the quality of software products.

The respondents mentioned different aspects in which improvement is desired. Sixty percent (60%) of the respondents stated that delivering software products on-time should be improved as most of the software products are delivered after the agreed deadline. Many respondents stated the need to consider the system architecture (macro view) without merely focusing on individual software components (micro view) in order to ensure the smooth functioning of the whole system. Some respondents highlighted the importance of managing the external vendors collaboratively.

Managers highlighted the importance of documentation practices in software development, although it is given the least priority due to various constraints. Many managers highlighted that most software systems do not have backup systems due to various reasons, but the presence of back up and disaster recovery systems will enhance the quality of software systems. Respondents mentioned that most of the software products do not match customer expectations, i.e., the end product is not what is expected by the customer. This is mainly due to the low level of interaction with customers. Further, problems of software defects were highlighted as a major complaint with regard to software quality. Some systems had poor user-interface and application security levels which led to customer dissatisfaction.

Respondents mentioned several factors to ensure better software quality but all those factors could be categorized under technological excellence, process maturity, team productivity or stakeholder involvement. However, not all respondents mentioned all four key factors. For example, a manager at JKCS (John Keells Computer Services) mentioned technology, resource skills, understanding of domain knowledge, expertise of testers and regular code reviews as factors for better software quality. These factors could be categorized under technological excellence (technology), team productivity (resource skills), and process maturity (understanding of domain knowledge, expertise

of testers, regular code reviews). It was mentioned that technology developments are quickly adopted for software development, and open source development frameworks have evolved rapidly. Many managers stated that due consideration should be given to business analysis, software engineering, quality assurance and project management processes.

With regard to reliability of software products, respondents mentioned that it was neither very high nor very low as a certain number of bugs are found in most of the software applications. Further, it was mentioned that transaction volume was high in most of the applications but there were system failures due to high transaction volumes. Under the application security aspect, respondents highlighted that it was an area of concern as there were many reported instances of system hacking and unauthorized system access. Respondents stated that portability is generally at a high level mainly due to the technological developments as many programming languages are platform-independent. It was highlighted that due consideration had been given to usability aspects although there were well-known products which have poor graphical interfaces. However, most of the software products did not have up-to-date user manuals. Respondents mentioned that availability of software systems was at an accepted level, but none of them stated it as 100%, which indicates that there is scope for improvement in product quality.

11. Discussion of Findings

The quantitative and qualitative information gathered from questionnaires and structured interviews are summarized below under each variable.

Technological Excellence

There is a positive relationship between technological excellence and software product quality and its strength is moderate (Pearson correlation +0.432). In most of the interviews with managers involved in software development, it was revealed that technology had become a standard and that there was no major difference in technology among companies. In other words, new technological developments are quickly adopted in software development. In an interview the general manager of a leading company mentioned that software projects fail not due to competency issues of employees but due to the selection of outdated or inappropriate technology. Some managers mentioned that software developers preferred to use open source software tools more than closed source software tools, as open source tools are evolving at a faster rate. Some respondents stated that software developers focus more on technology implementation compared to technology design. Thus, it was highlighted that greater attention should be given to technology design aspects such as database design, interaction diagrams, object oriented designs, etc. Since software development is technical, it is difficult to compete with outdated technology. In order to survive, it is necessary to upgrade to new technology quickly. Software should be backed by appropriate hardware. The performance of software will be greatly affected by the

performance of hardware as software works on the top of hardware. Some of the respondents mentioned that software product quality was high in the current context compared to past years mainly due to the development of software development frameworks and tools. For example, respondents stated that Microsoft.NET and Java based software development frameworks and tools have evolved rapidly in the past few years leading to development of high quality software products.

Therefore, quantitative and qualitative information indicates that high technological excellence leads to high software product quality.

Process Maturity

There is a positive moderate relationship between process maturity and software product quality (Pearson correlation +0.600). Software development is concerned with four main processes, viz., business analysis, programming, quality assurance and project management.

Business analysis is mainly concerned with gathering user requirements and preparing functional specifications. Most of the respondents have mentioned that the business analysis function is critical as all depends on how well the requirements were captured at the beginning. If the requirements were captured incorrectly, the end product would not satisfy customer needs. In order to build a high quality software product, it is necessary to identify user requirements thoroughly, and quality should be built from the beginning. This is in line with the study carried out by Fenton and Neil in 1999. Further, it was mentioned that upon identification of user requirements, prototypes should be built and demonstrated to the customer before the development of the actual software product. This will help clarify requirements better, and avoid any misunderstanding between customer and software development professionals. Also, it was mentioned that domain knowledge of functional consultants was very important in software development. One of the main barriers in this regard is the lack of training and development in specialized fields. Therefore, maturity of the business analysis process is vital to ensure high quality software products.

Actual development of the product occurs in programming where software engineers develop the product according to the requirements identified earlier. Respondents have mentioned that the skill level of software developers plays a key role in delivering high quality products.

The quality assurance process is vital to ensure that the developed software products are tested for quality. Most of the respondents mentioned the importance of having a separate quality assurance process as in many companies it is carried out either by software engineers or business analysts. The presence of a specialized quality assurance process will ensure that software products are viewed from an angle that is different to that of software engineers and business analysts. This is vital in ensuring that the

customer gets a product of high quality. Even some companies that were surveyed have adopted automated software product testing via tools such as HP Quality Center where requirements, product functions and test cases are linked together.

Many respondents stated the importance of having a proper project management function, since software development is carried out in projects, and management of these projects is critical. It was revealed that most of the software projects failed due to the lack of a professional project management process, since it is the process which oversees all other processes. Therefore, due consideration should be given for project management process in order to build high quality software products.

The quantitative and qualitative information shows that when maturity of software development process is high, it increases the quality of software products.

Team Productivity

There is a positive moderate relationship between team productivity and software product quality (Pearson correlation +0.553). Even with the world's best technology and processes, products will not be of high quality if there is low productivity among software development teams. One of the main findings is that both efficiency and effectiveness are not at a high level in most software development teams. It was stated that inability to execute timely product delivery has affected the relationship with clients. Further, delays in software projects have resulted in monetary losses as well.

Composition of the software development teams and the context within which teams operate are two important aspects in ensuring high product quality. It was found that knowledge and skill levels of team members play an important role in software product quality. This is in line with the findings of Krishnan (1998), where it was identified that higher levels of domain and language experience lead to high software product quality. It was recognized that most of the software development teams operate in an informal culture and a flat organizational hierarchy. Some of the respondents in software development units mentioned that team members were not performing up to expectations due to various issues such as monotonous work and lack of task significance. However, most of the teams have taken steps to overcome those issues and improve productivity. For example, they have started rotating the employees among different software projects to ensure that team members work on different product domains and technologies. This will also help in the career development of software professionals. The quality assurance manager of a software development team stated that more attention should be paid to the development of job descriptions of team members.

Leadership style plays an important role in managing team productivity. Most of the team leaders follow a participatory leadership style and there is not much distance between the leader and subordinates. It was identified that software development teams work according to flex time. A manager of a software development team stated that they had team activities every Friday evening for half an hour. When team members

participate in team building activities, it improves communication among team members, make the workplace more enjoyable and motivate the team. It also helps to identify and utilize the strengths of team members and this will lead to improved productivity.

As discussed earlier, quantitative and qualitative information indicates that high team productivity leads to high software product quality.

Stakeholder Involvement

There is a positive moderate relationship between stakeholder involvement and software product quality (Pearson correlation +0.617). Stakeholder involvement is critical for software product development. Interaction with the customer and identification of customer needs are very important. It was identified that most of the software products are of low quality since software professionals do not interact with the customers frequently. However, software companies have taken steps to conduct interim product demonstrations so that feedback could be obtained from customers before producing the final product. This has contributed to minimizing change requests as well as unnecessary costs.

Further, management support is vital for software development. It is necessary to ensure that adequate budgets and resources are allocated for software development projects. Most of the respondents mentioned that they were motivated when there WAs management support to execute work. But it was identified that goals and objectives of software projects have not been properly communicated to the software development teams by the management. Therefore, management of software development companies and units has not devoted adequate attention to the critical processes compared to non-critical processes. These findings are in line with the study carried out by Zwikael (2008) in Japan, Israel and New Zealand.

Vendor engagement is necessary to ensure that external vendors provide the necessary equipment and resources to deliver high quality software products. It was identified that companies had failed to deliver high quality software because they had selected partners who were not very cooperative. Therefore, maintaining a high level of engagement with external vendors will lead to the delivery of high quality of software products.

The foregoing account shows that higher stakeholder involvement in software development projects will lead to higher software product quality.

12. Conclusions and Recommendations

The software industry has become a strategically important industry for Sri Lanka. It has the potential to contribute to Sri Lanka's exports. On the one hand, it is less capital-intensive and on the other, the relatively low-cost, skilled labour can attract many

software related investments. Meanwhile, India is competing strongly and has been able to attract many software investments. Facing this competition needs enhancement of software product quality in Sri Lanka.

Software products are being used by almost every industry and the importance of software product quality has increased rapidly over the past few years. As discussed previously, technological excellence, process maturity, team productivity and stakeholder involvement are positively related to software product quality. Therefore, an increase in one or more of these factors will increase software product quality. When the coefficient of determination is calculated, taking all variables into consideration, it shows that the four independent variables together explain a high variation in the dependent variable. Therefore, it will be necessary to improve the existing level of these factors affecting software product quality.

The following recommendations are made for the improvement of software product quality.

Technological Excellence

Software teams must improve software designs through the use of the latest design methods, and the use of software design and development tools will give an integrated view of all software components. It would be required to ensure hardware of adequate capacity, memory and processing power before the commercial launch of software products. The availability of disaster recovery systems is required in real time processing systems, in order to minimize the impact of system downtime.

Process Maturity

Conducting more requirement gathering meetings with customers and development of prototypes to validate requirements will assist in identifying customer requirements at the beginning. Regular code reviews and adherence to coding standards are important in maintaining software programmes. Further, keeping updated documentation such as requirement specification, functional specification, technical specification, release notes, project plans and implementation plans is vital for software product quality. The increase in the depth and breadth of software testing along with test automation tools will ensure that software products have less errors. Therefore, effective alignment between social (software professionals) and technical systems (technical resources) is mandatory. From a project management perspective, planning and controlling the software projects and monitoring project progress throughout the life cycle is needed to deliver high quality software products.

Team Productivity

Assigning software team members to different projects exposes them to different product environments. It is necessary to create the climate for team work and conduct team building activities. Further, software teams must be equipped with adequate

resources. It is necessary to increase the motivation levels of team members *via* financial and non-financial rewards. Team management must create the opportunity for training and development in order to acquire new knowledge as the software industry is growing fast.

Stakeholder Involvement

Frequent interactions with customers during the product development life cycle will ensure that the ultimate product meets customer expectations. Further, specific objectives must be set for software projects with adequate budgets and realistic timelines. Selection of suitable vendor partners through a proper evaluation scheme is recommended as the quality of external supplies affects the quality of software products.

References

A software quality crisis is brewing (2008). Retrieved 7th July 2009, From http://searchsoftwarequality.techtarget.com/news/column/0,294698, sid92_gci1333425, 00.html

Abreu, F.B. and Melo, W. (1996). Evaluating the impact of object oriented design on software quality, Retrieved 8th June 2009, From http://www2.computer.org/portal/web/csdl/abs/proceedings/metrics/1996/7364/00/73640090abs.htm

Acuña, T., Gomez, M. & Juristo, N. (2008). Towards understanding the relationship between team climate and software quality-a quasi-experimental study, Empirical Software Engineering, 13, No. 4, pp. 401-434

Burns, H.S. and Halliburton, R.A. (1989). Tackling productivity and quality through customer involvement and software technology, Global Telecommunications Conference, IEEE Vol. 1, pp.631- 635

Closed source software (2009). Retrieved 27th July 2009, From http://en.wikipedia.org/wiki/Closed_source

Diaz, M. and Sligo, J. (1997). How software process improvement helped Motorola, IEEE Software, Vol. 14, No. 5, pp. 75-81.

Glossary (2009). Retrieved 27th May 2009, From http://thiyagarajan.wordpress.com/glossary/

Harter, D.E. and Slaughter, S.A. (2000). Process maturity and software quality: a field study, International Conference on Information Systems, Brisbane, Queensland, Australia, pp. 407-411

ICT Profile - Sri Lanka (2006). Retrieved 27th May 2009, From http://www.apdip.net/projects/dig-rev/info/lk/ Kang, H. et al. (2006). Factors in team effectiveness: Cognitive and demographic similarities of software development team members, Retrieved 8th August 2009, From http://hum.sagepub.com/ cgi/content/abstract/59/12/1681