SOFTWARE QUALITY AND CERTIFICATION: PERCEPTION AND PRACTICES IN MALAYSIA

*Jamaiah Haji Yahaya, **Aziz Deraman, and **Abdul Razak Hamdan

**Faculty of Information Technology,*
Northern University of Malaysia, Sintok, 06010, Kedah, Malaysia

***Faculty of Information Science and Technology,*
National University of Malaysia, 43600 Bangi, Selangor, Malaysia

jamaiah@uum.edu.my

ABSTRACT

Recently, there has been an increasing amount of literature on software quality and at the same time complaints are reported widely on the quality status of software products. In the past few years, we have seen a tremendous growth of companies and agencies involved in the software industries not only in Malaysia but also in the entire world. The emergence of Multimedia Super Corridor (MSC) in Malaysia in 1996 was the starting point for the blooming of software and ICT related companies. Despite that, not much attention is given to the quality of software product that is being developed by various categories of companies. These companies could not justify the quality of their products to users and these users are left with uncertainties on the quality of software. This study was conducted to investigate the current practice and perception on software quality and assessment in Malaysian industries. A total of 43 organizations were surveyed to answer several issues on quality and certification. Some of the important findings from this survey were the need for standard and mechanism in software assessment, and the need for independent party certification to justify the quality status of software products. This paper discusses more findings of this survey.

Keywords: Software quality, software certification, Malaysia.

1.0 INTRODUCTION

The Multimedia Super Corridor (MSC) is Malaysia's most exciting initiative for the global information and technology (ICT) industry. It has been conceptualized for about 10 years since 1996 and hosts hundreds of multinational, foreign-owned, and home-grown Malaysian companies. These companies focus on multimedia and communications products, solutions, services, and research and development. In addition to the MSC companies, there are growing numbers of software house companies spread all over Malaysia. These companies are involved in various types of ICT services and some are focused on software development. Some of these companies are providing services as out-source companies to government agencies in software development and maintenance. Government and semi-government agencies develop software in-house with their own ICT personnel as well as purchasing software from vendors.

Whittaker and Voas (2002) discussed the topic of software quality from the very early days of software development (1950s) to the current era of software development (2000s). They claimed that software quality is no better today than it was decades ago. In some cases, it is worse (Voas, 2000). There are several other studies on software quality such as by Punter, Solinger, and Trienekens, (1997), Leung (2001), and Van Zeist and Hendriks (1996). Three factors have been considered for determining quality of software, which are product, development processes, and personnel. Voas (1998) referred this as a triangle of software quality. This research focuses on determining quality through end product approach.

Specifically, this paper focuses on observing the current perceptions and practices in assessing software product quality and certification in Malaysia. The importance of this study lies in the growing of interest among organizations and companies in obtaining software product certification, as well as the lack of mechanisms for doing so.

2.0 SOFTWARE QUALITY

Quality in the software context means involving a variety of quality attributes, for example performance, security, reliability, and other attributes. International Organization for Standardization (ISO) defined quality as the totality of features and characteristics of a product or services that bear on its ability to satisfy stated or implied needs (Yamada, 1996). Generally, people think of quality as conformance and compliance to specification continuously and consistently. IEEE defines software quality as a software feature or characteristic used to assess the quality of a system or component (IEEE, 1993).

Literature on this subject covers several software quality models from McCall, Boehm, FURPS, IS09126, Dromey and Systemic. The McCall quality model (Rae, Robert, & Hausen, 1995; Norman & Pfleeger, 1997) is one of the earliest models and commonly called the Factor Criteria Metric (FCM) model. This model is usually constructed in a tree-like fashion. The upper branches hold important high-level quality attributes, such as reliability and usability that need to be quantified. Each quality attribute is composed of lower-level criteria. The factors included in this model are: correctness, reliability, efficiency, integrity, usability, maintainability, testability, flexibility, portability, reusability, and interoperability. One aspect which was not considered directly in this model is functionality.

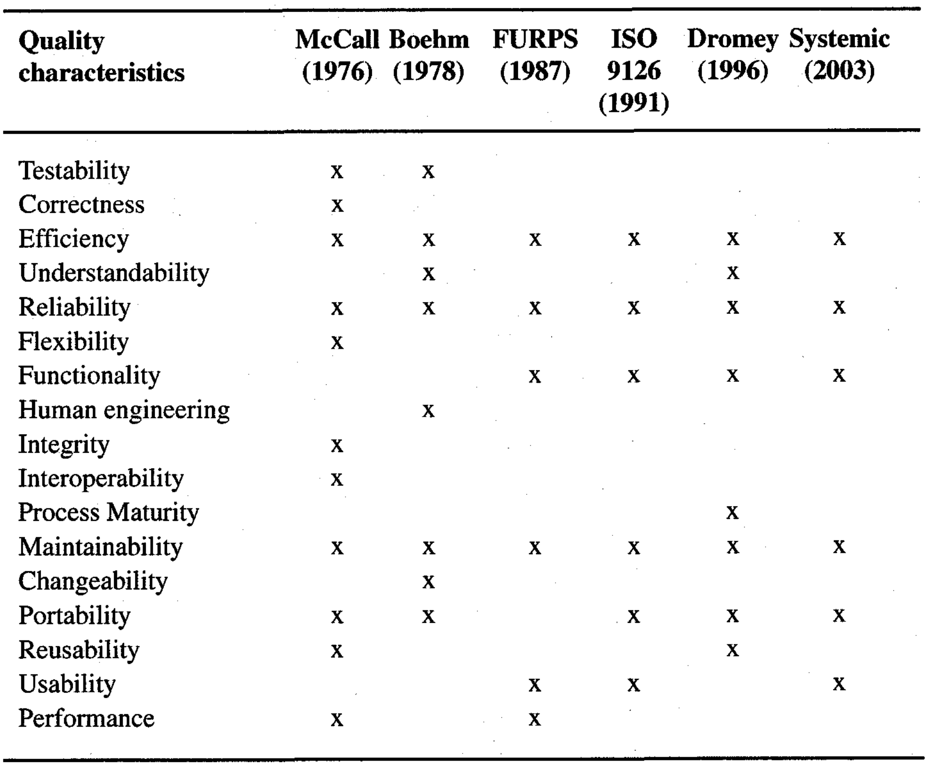
The Boehm model (Khosravi & Gueheneuc, 2004; Norman & Pfleeger, 1997) is similar to the McCall model in that it represents a hierarchical structure of characteristics, each of which contributes to total quality. The Boehm model views software with general utility. General utility is then broken down into portability, utility, and maintainability. Utility is further broken down into reliability, efficiency, and human engineering. Maintainability is in tum broken down into testability, understandability, and modifiability. This model is presented in levels called primary uses, intermediate construct, and primitive constructs.

The FURPS model takes five characteristics of quality attributes and that make up its name: Functionality, Usability, Reliability, Performance, and Supportability. When the FURPS model is used, two steps are considered: setting priorities and defining quality attributes that can be measured (Khosravi & Gueheneuc, 2004). One disadvantage ofthis model is that it does not take into account the software product's portability (Ortega, Perez, & Rojas, 2003).

ISO 9126 defined product quality as a set of product characteristics. The characteristics that govern how the product works in its environment are called external quality characteristics, while the characteristics relating to how the product is developed are called internal quality characteristics. ISO 9126 indicated six main quality characteristics which are associated with several subcharacteristics (Yamada, 1996). The characteristics are efficiency, functionality, maintainability, portability, reliability, and usability. One advantage of this model is that it identifies the internal characteristics and external quality characteristics of a software product. However, at the same time it has the disadvantage of not showing clearly how these aspects can be measured (Rae et al., 1995).

Dromey (1996) proposed a working framework for building and using a practical quality model to evaluate requirement determination, design, and implementation phases. This information can be used directly to build, compare, and evaluate better quality software products. In comparing with ISO 9126, additional characteristics such as process maturity and reusability are noticeable. Process maturity is an attribute that has not been considered in the previous models.

Table 1: Quality characteristics present in the different models: McCall, Boehm, FURPS, ISO 9126, Dromey, and Systemic



Quality characteristics	McCall (1976)	Boehm (1978)	FURPS (1987)	ISO 9126 (1991)	Dromey (1996)	Systemic (2003)
Testability	x	x				
Correctness	x					
Efficiency	x	x	x	x	x	x
Understandability		x			x	
Reliability	x	x	x	x	x	x
Flexibility	x					
Functionality			x	x	x	x
Human engineering		x				
Integrity	x					
Interoperability	x					
Process Maturity					x	
Maintainability	x	x	x	x	x	x
Changeability		x				
Portability	x	x		x	x	x
Reusability	x				x	
Usability			x	x		x
Performance	x		x			

Systemic model is a model that takes into account the quality of product and process. It considers the characteristics in the ISO 9126 and Dromey model (Ortega et al., 2003).

Table 1 summarises the quality characteristics identified in different models from McCall, Boehm, FURPS, IS09l26, Dromey, and Systemic in chronological order of their appearance. It shows that the main quality characteristics found in the majority of the models are: efficiency, reliability, maintainability, portability, usability, and functionality, which are presented in more recent models. These characteristics appear in all models and therefore, are considered as essential and vital.

3.0 SOFTWARE CERTIFICATION

The term certification in general is defined as "a written testimony or voucher, especially of character or ability" (Rae et al., 1995). A software certification is defined by Jeffry Voas as a fact sheet that spells out known software output behaviours (and it could also spell out known internal behaviours). It also spells out what conditions those behaviours can manifest themselves (Voas, 1999a). Stanfford and Wallnau (1997) defined certification as a process of verifying a property value associated with something, and providing a certificate to be used as proof of validity. Certification is a means for improving the discipline by promoting the practical implementation of standards, the awareness of a body of knowledge, the recognition of a code of ethics, and the need for professional development (Tripp, 2002).

With the development of software certification, users might be able to choose the correct software that meets their requirement even though users do not understand the processes and program underlying the complete software. The paper by Aldrich, Goulde, and Wong (2000) discussed the findings on the importance of certification in terms of cost reduction. One way to conduct software certification is through involvement of end users in the process. In this approach, the independent certification body collects valuable information from the user's environment and collects on how the product is used (Voas, 1999b).

Certification does not assert that the software is correct, but instead provides some levels of confidence to the users in terms of products' quality. Therefore, a mechanism and standard for certifying software products is required. The certification process is to be implemented by a team that is independent from the developers of software.

4.0 RESEARCH METHODOLOGY

We performed a survey to investigate the current practice and perceptions on assessment of software products and to determine the need for software quality standards in the Malaysian industry. The instrument used to gather the data is through mail questionnaires and interviews. There are 54 items in the survey form which are divided into four main sections. The first section is on the respondent background, the second section is the organizational background, third section is product quality control, and the last section is on human resource development. The product quality control section consisted of software acquisition, quality factors, software quality assurance, and software acquisition policy.

1. Software acquisition

   In this part of the questionnaire, we investigated current practice of software acquisition process in Malaysia. The first question asked was:

   Is it useful to have information on software acquisition, submission and termination by third party before you purchase a product? The scale used is a five point Likert scale 1 Where 1 represents *not useful,* 2 refers to *low,* 3 refers to *average,* 4 represents *useful,* and 5 represents *very useful.* The second question was: In general, what is your overall evaluation on the software that already available in your organization? The scale used here is a five point Likert scale where 1 refers to *not acceptable,* 2 refers to *acceptable*, 3 refers to *average,* 4 represents *good,* and 5 represents *excellent.*

2. Quality Attributes

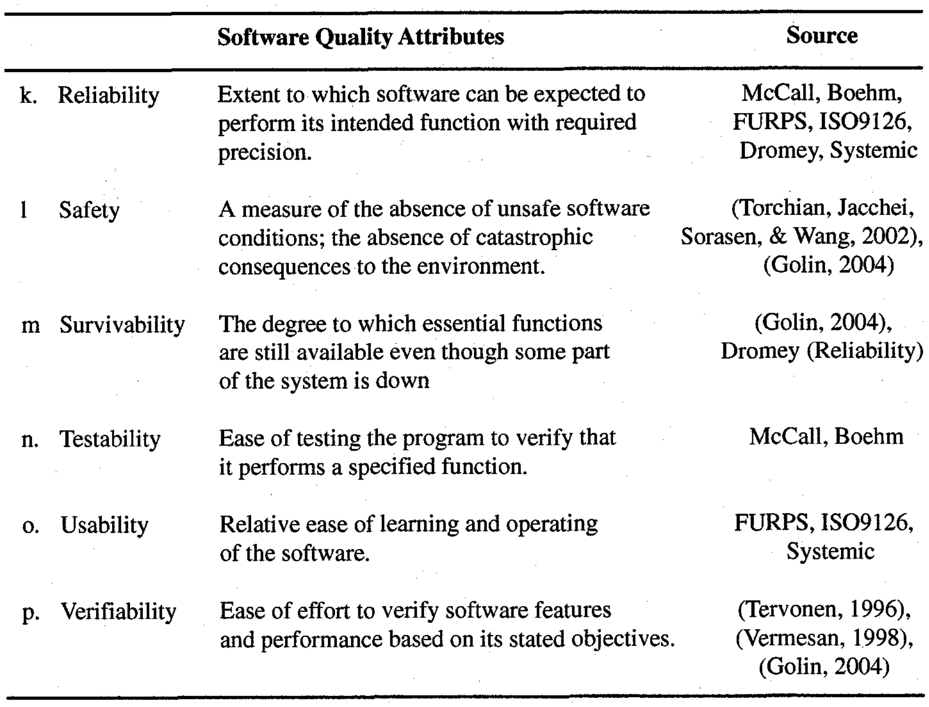
   The quality attributes or factors are identified from literature study. The attributes listed in the survey form are considered vital and commonly used in many software assessments. The attributes are efficiency, expandability, flexibility, functionality, integrity, interoperability, intraoperability, maintainability, portability, reusability, reliability, safety, survivability, testability, usability, and verifiability. The description of the attributes is shown in Table 2.

   Respondents were asked to rank the level of consideration of the listed quality factors before selecting and purchasing or accepting and installing any software products. The rank is according to a Likert scale given as not considered (1), low consideration (2), average (3), high consideration (4) and very high consideration (5).

Table 2: Attributes of software quality

		Software Quality Attributes	Source
a.	Efficiency	Extent to which the software is able to do more with less system (hardware, operating system, communications, etc.) resource.	McCall, Boehm, FURPS, ISO9126, Dromey, Systemic
b.	Expandability	Relative effort required for expanding software capabilities and/or performance by enhancing current functions or by adding new functionality.	McCall (Maintainability), (Golin, 2004)
c.	Flexibility	Ease of effort for changing the software's mission, functions or data to meet changing needs and requirements.	McCall
d.	Functionality	The capability of the software to provide functions which meet stated and implied needs when the software is used under specified conditions.	FURPS, ISO9126, Dromey, Systemic
e.	Integrity	Extent to which access to software or data by unauthorized users can be controlled.	McCall
f.	Interoperability	Relative effort needed to couple the software on one platform to another and/or another platform.	McCall
g.	Intra-operability	Effort required for communications between components in the same software system.	(Murine, 1983)
h.	Maintainability	Ease of effort for locating and fixing a software failure within a specified time period.	McCall, Boehm, FURPS, ISO9126, Dromey, Systemic
i.	Portability	Ease of effort to transfer software from one hardware configuration and/or software system environment to another.	McCall, Boehm, ISO9126, Dromey, Systemic
j.	Reusability	Ease of effort to use the software (or its components) in another software systems and applications.	McCall, Dromey,

(continued Table 2)



	Software Quality Attributes	Source
k. Reliability	Extent to which software can be expected to perform its intended function with required precision.	McCall, Boehm, FURPS, ISO9126, Dromey, Systemic
l Safety	A measure of the absence of unsafe software conditions; the absence of catastrophic consequences to the environment.	(Torchian, Jacchei, Sorasen, & Wang, 2002), (Golin, 2004)
m Survivability	The degree to which essential functions are still available even though some part of the system is down	(Golin, 2004), Dromey (Reliability)
n. Testability	Ease of testing the program to verify that it performs a specified function.	McCall, Boehm
o. Usability	Relative ease of learning and operating of the software.	FURPS, ISO9126, Systemic
p. Verifiability	Ease of effort to verify software features and performance based on its stated objectives.	(Tervonen, 1996), (Vermesan, 1998), (Golin, 2004)

3. Software quality assurance

   This section would like to investigate the activities of Software Quality Assurance (SQA) team in organizations in Malaysia - the management commitment. Several questions related to these subjects were asked and respondents had to tick suitable answers from the lists.

4. Software acquisition policy

   This part of the questionnaire intended to study the demand for standard and certification in the software industry. Respondents replied to related questions by ticking the correct and suitable answer from the list.

A total of 140 questionnaires were distributed to various organizations and industries in Malaysia. The sampling frame for this survey was a listing of organizations and companies that owned homepages and have links to the ministry websites, statutory body links, Malaysia Super Corridor links, and other links such as Malaysia Central, Muslim Trade Network, and Reference Directory. We were interested to investigate companies that have homepages on the Internet to be selected and participate in this survey.

The sampling technique used was simple random sampling. A total of 43 respondents had responded to the survey. Each respondent represented his or her organization. Participants were selected from the upper management and middle management, and also project leaders, software engineers, researchers, consultants, IT professionals, and financial officers. They are those involved directly or indirectly in software acquisition in any of the following: standardization and quality process, software development, purchases of product and services authorization, budget for development initiative, recommendation on strategic direction, and provider selection.

5.0 DATA ANALYSIS

The analysis was done using a statistical package. The first section discusses the demographic and general information on the respondents in this survey. The following sections' discuss the result of the analysis.

5.1 Distribution of the category of respondents

The percentage of completed and returned questionnaires is 31 %. The respondents come from government agencies (26%), semi-government agencies (30%), and private sector (44%).

5.2 Who are the respondents?

Fig. 1 displays the distribution of respondent's position in this survey. The survey shows that 35% of the respondents were in middle management, 21 % were in upper management, and 16% were consultants. This is followed by software engineers (12%), project leaders (7%), researchers (5%) and financial officers and IT professionals (2%).

The majority of respondents in the survey (37%) have work experience between three to 10 years in the organization. About 28% of respondents have work experience of 11 to 20 years and 19% have work experience of more than 20 years in the organization. Only 6% of respondents have work experience of less than three years.

5.3 Distribution of organization function

The greatest number of respondents is from the public administration and services (27%), followed by the education/training sector (25%) (see Fig. 2).

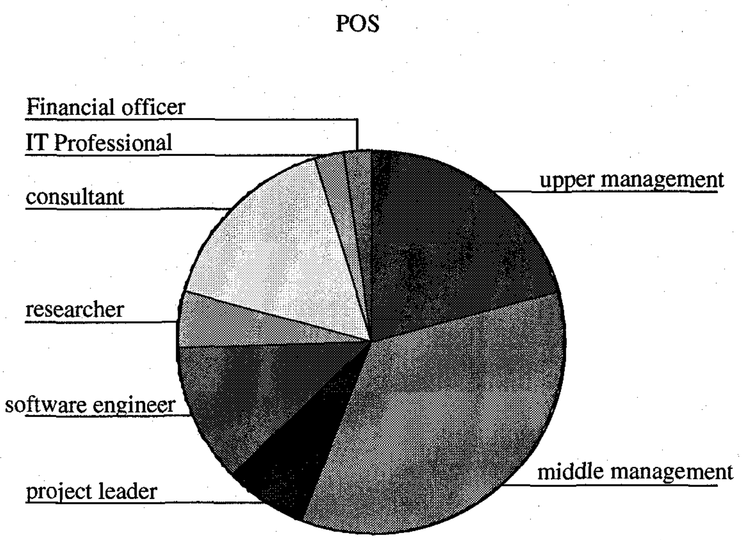


Fig. 1: Distribution of respondents

These groups are followed by softwarelhardware (18%) and banking/finance/ trust (11 %). Over 80% of our respondents were employed by organizations in these four sectors. Smaller groups came from healthcare, agriculture, logistics/ construction, and other categories.

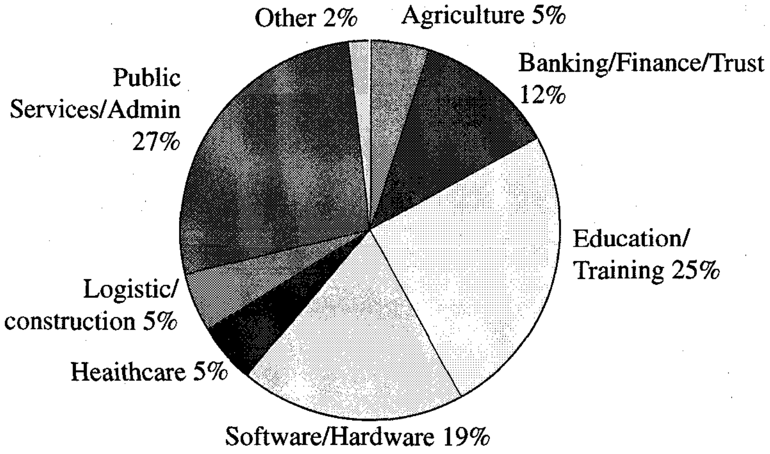


Fig. 2: Category of respondents by industry

5.4 Number of employees in the organisation

The analysis indicated that 49% of the respondents were from big companies, which have more than 1000 employees. About 30% of respondents were medium size companies with 100 to 1000 employees, and 21 % were small companies with less than 100 employees.

5.5 Number of organizations with or without special IT unit/department

The analysis shows that 88% of the respondents had special IT department in the organization and only 12% had no special IT department in their organisation. A substantial portion of (93%) organizations were involved in software development or customization while only 7% were not involved in these activities.

6.0 FINDINGS AND DISCUSSION

6.1 Current practice in software product acquisition, criteria of selection

Respondents were asked to indicate the important criteria for software selection from the list of given criteria in the survey form. The selected criteria by respondents were the current practices in their organizations.

The results showed that main criteria of software selection by all respondents were cost (74.4%), followed by organizational requirement (69.8%), quality (62.8%), and ease of use (46.5%). This seems to indicate thatthese four criteria are the main consideration before selecting or purchasing software in organizations in Malaysia. The analysis done by type of organizations which are government agencies, semi-government agencies, and private sectors shows slightly different results. For example, the private sector indicated that the important criteria are cost, followed by organizational requirement, and quality, whereas for government agencies, the main criteria are organizational requirement, followed by quality and cost (see Table 2).

6.2 Perceptions on existing software in the organizations

In evaluating the quality of existing software in organizations in Malaysia, it was observed that the quality of the software is still uncertain with various levels of assessment. We asked our respondents to evaluate the existing software in terms of quality based on their perceptions. The scales for this evaluation were; not acceptable, acceptable, average, good, and excellent. In general, it was observed that 60% of respondents agreed that available software in their organizations is rated good and excellent, while 40% agreed that the software is rated on the scale of average, acceptable, and not acceptable.

Table 2: Current practice in software selection

Criteria of selection	Type of Organisation			Overall Respondents (%)
	Government (%)	Semi-Government (%)	Private Sector (%)	
Quality	73	69	53	62.8
Popularity	0	15	16	11.6
Ease of use	64	46	37	46.5
Cost	66	69	84	74.4
Familiarity	9	0	5	4.7
Human resource aspect	27	23	10	18.6
Organisational requirement	73	69	68	69.8
Policy	0	8	10	7.0
De facto standard	0	8	10	7.0
Vendor recommendation	18	8	5	9.3
Users recommendation	9	7	15	11.6

6.3 Software quality attributes in product assessment

The following analysis is to determine the main quality attributes in software product assessment by respondents. We asked respondents to distinguish the importance of each of them. The survey indicated that functionality, efficiency, integrity, maintainability, and reliability were the main attributes with high and very high consideration in assessing software products by all respondents of this survey. We established the result by assuming and counting that the mean score of 4 (with high consideration) and above. The mean score given by the respondents were as follows: functionality = 4.33, efficiency = 4.05, integrity = 4.23, maintainability = 4.12, and reliability = 4.33. Usability and flexibility were rated a little bit lower by respondents, with a mean score of 3.95. Other attributes with lower mean scores were considered as not commonly used with less consideration in assessing software products in organizations in the Malaysian industry. The detailed results are shown in Table 3.

Table 3: Software quality attributes: means score by type of organization

Software Quality	Type of organization			All
Attributes	Government	Semi-govern	Private Sector	Respondents
Functionality	4.09	4.62	4.26	**4.33**
Efficiency	3.73	3.92	4.32	**4.05**
Expandability	3.82	3.92	3.89	3.88
Flexibility	4.09	3.85	3.95	**3.95**
Integrity	4.27	4.23	4.21	**4.23**
Inter-operability	3.73	4.23	3.68	3.86
Intra-operability	3.91	4.00	3.79	3.88
Maintainability	4.00	4.15	4.16	**4.12**
Portability	3.73	3.69	3.74	3.72
Reusability	3.91	3.31	3.37	3.49
Reliability	3.82	4.46	4.53	**4.33**
Safety	4.00	3.62	4.00	3.88
Survivability	3.55	3.46	3.68	3.58
Testability	3.64	3.77	3.74	3.72
Usability	3.91	3.85	4.05	**3.95**
Verifiability	3.73	3.54	3.89	3.74

Table 4: Comparison between ISO 9126 model and survey result

Quality Attributes	ISO 9126	Survey
Efficiency	×	×
Reliability	×	×
Functionality	×	×
Integrity		×
Maintainability	×	×
Portability	×	
Usability	×	

The comparison between the ISO 9126 model and the result from the survey is shown in Table 4. The ISO 9126 model includes efficiency, reliability, functionality, maintainability, portability, and usability, and many researchers use this model as their base line of their research. However, our survey concluded that the main attributes for software quality and assessment are efficiency, reliability, functionality, integrity, and maintainability. Our survey also indicated that usability is slightly lower in terms of importance in the assessment.

6.4 Software Quality Assurance

The survey indicated that usually, respondents test the product themselves before deciding to acquire any product. The testing of the software was usually done by the SQA team in the organisations. We asked the respondents: "In most cases, software testing by itself is not sufficient to establish confidence that the software is fit for its intended use. Do you agree?" About 88% agreed while 11 % disagreed. In order to investigate the influence of the SQA team in testing, the survey showed that 81 % of respondents with an SQA team agreed with that statement and 19% disagreed. However, for respondents without an SQA team, the percentage is higher; 96% of this group agreed and only 4% disagreed. Therefore, this result shows that an independent certification party may solve some of the problems in quality since the SQA department itself does not guarantee the quality of the product. The ideas of certification by an independent party have been discussed in several papers (Voas 1998; Voas 1999b; Voas 1999c; Maginnis 1999; Schneidewind & Norman, 1996).

6.5 Decision making on software product assessment

Before any decision for selecting and purchasing a product, the majority of respondents (79%) tested the product themselves, 62.8% relied upon recommendation from other users, 37% depended on a report from third party, and 37% depended on recommendation by the vendor. The results showed that besides testing the products themselves, respondents depend quite highly upon recommendation from other users and friends. The recommendations may be influenced by attitude, culture, and unofficial reports from other parties and friends. The reports and recommendation may be biased to certain products.

Whilst many respondents (51.2%) believed and indicated that good quality software will cost more, almost one-third of the respondents (34.9%) believed that it will cost more at the beginning but it will reduce in the long-term.

6.6 User's awareness on software certification

We ask our respondents whether they have heard of anything about software certification, and 81 % answered *Yes* and 19% answered *No.* The majority of them (77%) agreed that certification would assure that the product complies and adheres to software requirements and standards. This also shows the trustworthiness of third-party certification by respondents. The respondents agreed with the trustworthiness of the independent body to validate and certify their products. However, respondents indicated that the characteristics of the independent body is very important and reflect the trustworthiness of the body. Therefore, the ethical matter surrounding certification is very important to resolve. There is a discussion on this issue by Miller and Voas (1999).

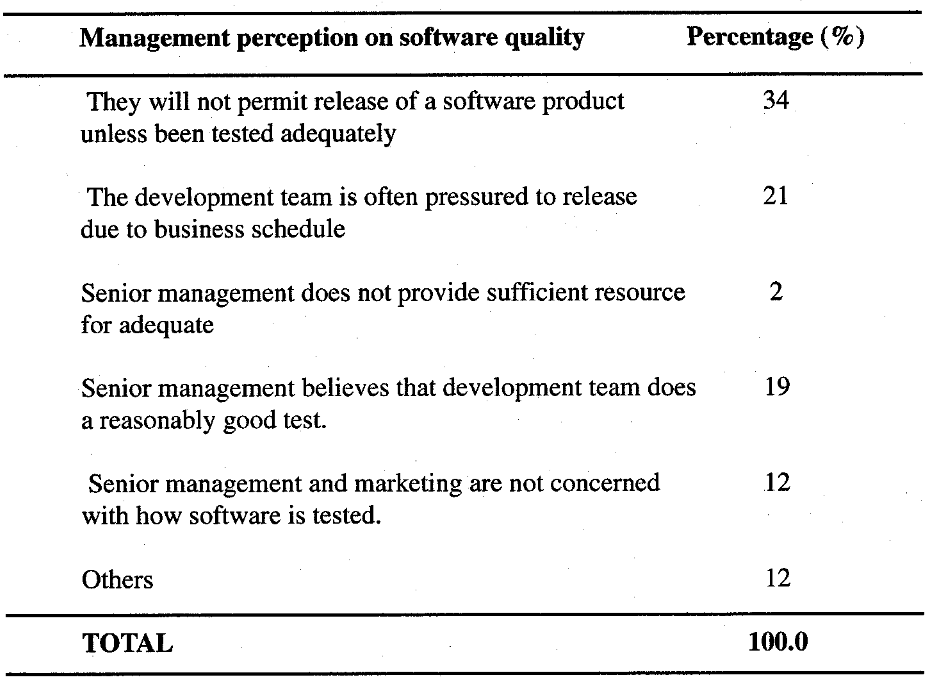
The following analysis is to determine that standards and certification needed and demanded by the industry in the near future. Respondents were asked to indicate how likely were they to choose certified software given a choice between two similar software. The survey indicated that 53% of the respondents believe approximately twice as likely to choose certified software. Whilst 40% of the respondents will as likely to choose certified program, 7.0% indicate other choice. This result indicates that software with certification will be more in demand and have a market in the near future. There is a trend in buying software that gives preference for certified software.

This survey also indicated that 86% of the respondents thought that software certification assures that objective evidence exists prior to its use and it performs required functions. Only 14% do not agree. About 79% of the respondents agree that certified software has resource savings and 21 % do not agree with that statement. It is a clear indication that users give greater value for certified software. These findings are consistent with other surveys done by the Patricia Seybold Group in year 2000 (Aldrich et al., 2000).

6.7 Top management commitment in ensuring software quality

Whilst 76% of the respondents indicated that top management gives commitment in ensuring quality in software development and acquirement, 24% stated the opposite. About 34% indicated that the management will not permit the release of a software product unless it had been tested adequately, while 21 % indicated that the development team is often pressured to release software due to business schedule. Table 5 demonstrates the full list of answers by respondents.

Table 5: Senior management perception on software quality and testing



Management perception on software quality	Percentage (%)
They will not permit release of a software product unless been tested adequately	34
The development team is often pressured to release due to business schedule	21
Senior management does not provide sufficient resource for adequate	2
Senior management believes that development team does a reasonably good test.	19
Senior management and marketing are not concerned with how software is tested.	12
Others	12
TOTAL	**100.0**

In order to investigate reasons for software deployment by respondents despite quality issues, three main reasons have been identified: -

- Problem did not appear severe enough to delay deployment (44%)
- Issues were not known at the time of deployment (32%)
- The developer would fix the problem after deployment (10%)

6.8 Contribution to knowledge and skill in software development

The survey respondents indicated that the main contribution to knowledge and skill in software development is experience (41 %), followed by training (35%), and formal education (24%). Whilst more than 70% of respondents said that they never notice any seminars or conferences on software quality held nearby, 30% had noticed, while 30% of respondents indicated that attending seminars or conferences will improve and enhance their knowledge and expertise in their area. About 58% claimed that seminars are useful in certain aspects only, while 5% of the respondents said that they just repeat what they already knew.

7.0 CONCLUSION

Software certification process may be categorized into 3 main aspects: process, product, and personnel. This research investigated the issues of quality and certification of software product. Results of this survey showed that organizations in Malaysia were aware of the issues of quality and certification and reasonably measure the quality aspects to obtain good software products in the organizations. Findings from the survey also showed that the demand for a software quality standard and a better mechanism to assess software are required. Thus, the value of software certification to the industries cannot be denied. Also, the survey does not reject the need for independent software assessment and certification in determining software quality.

Even though the majority of the researchers and projects done in Europe use the ISO quality model as their baseline of the research, we investigate the actual metrics relevant to the software product assessment in Malaysia. "If beauty is in the eye of the beholder, then quality must be as well" (Voas, 2004). There are several models of quality available from literature and we believe that quality is a complex concept because it means different things to different people since it is highly context dependent (Kitchenham & Pfleeger, 1996). Thus, there will be no single, simple measure of software quality acceptable to everyone. What we could do is to define the aspects of quality in which we are interested, and then decide how we are going to measure them. This survey shows that the most common factors for software quality assessment in the Malaysian industry are functionality, efficiency, integrity, maintainability, and reliability. Also included are usability and reliability with a little lower ranking of assessment. The identification of software quality attributes that are relevant to this environment will lead to the development of software product certification model and methodology.

In addition to the issues related to the development and implementation of software product certification model and methodology, higher learning institutions should also consider the importance of software quality in the academic curriculum. The significance here is the future demand for better software quality and standard by the market and users. Thus, future software practitioners should have an enhanced knowledge in software quality issues not only in the end product approach, but also in the development process
approach as well.

REFERENCES

Aldrich, S.E., Goulde, M.A., & Wong, F. (2000). *Customer wants windows 2000 certified applications: Certification delivers greater reliability and lower cost.* Retrieved on Jan 28, 2004, from http://www.etestinglabs.coml downloads/certification/cere benefits.pdf

Dromey, R. G. (1996). Cornering the chimera. *The IEEE Software,* 13(1), 3343.

Golin, D. (2004). *Software quality assurance: From theory to implementation.*
England: Addison Wesley.

IEEE (1993). *The IEEE standardfor a software quality metrics methodology.*
Retrieved on August 20, 2005, from http://ieeexplore.ieee.org/xpll standards.jsp

Khosravi, K & Gueheneuc, Y.G. (2004). *A quality model for design patterns.*
Retrieved on May 12,2005, from http://www.yann~ael.gueheneuc.netl *workfTutoringlDocumentsl* 041021 +Khosravi+ Technical-Report.doc.pdf

Kitchenham, B. & Pfleeger, S. L. (1996). Software quality: The elusive target.
The IEEE Software, 13(1), 12-21.

Leung, KN. (2001). Quality metrics for intranet applications. *Journal of Information* & *Management,* 38, 37-152.

Maginnis, P. T. (1999). *Linux certification for the software professional.*
Retrieved on January 13, 2005 from http://delivery/acm.org/1O.1145/ 330000/327737/a9-maginnis.html.

Miller, K & Voas, J. (1999). An ethical can of worms for software certifiers. *IT Professional, 1(5), 18-20.*

Murine, G.E. (1983). Improving management visibility through the use of software quality metric. *Proceedings of the IEEE Computer Society $ Seventh International Computer Software* & *Application Conference.* New York: Institute of Electrical and Electronic Engineers, Inc ..

Norman, EE. & Pfleeger, S.L. (1997). *Software metrics: A rigorous and practical approach,* Second Edition. Boston: PWS Publishing.

Ortega, M., Perez, M., & Rojas, T. (2003). Construction of a systemic quality model for evaluating a software product. *Software Quality Journal,* 11, 219-242.

Punter, **T,** Solingen, V. R., & Trienekens, J. (1997). Software product evaluation:
Current status and future needs for customers and industry. *Proceedings of the 4th Conference on Evaluation of Information Technology.*

Rae, A., Robert, P., & Hausen, H. (1995). *Software evaluation for certification:*
Principle, practice and legal liability. London: McGraw-Hill International.

Schneidewind, N. E & Norman E (1996). Do standards improve quality? *The IEEE Software,* 13(1),22-24.

Stanfford, J. & Wallnau, K.(1997). *Is third party certification necessary?*
Retrieved August 15, 2003 from http://citeseer.nj.nec.com/.

Tervonen.I. (1996). Support for quality-based design and inspection. *The IEEE Software,* 13(1),44-54.

Torchiano, M., Jaccheri, L., Sorensen, C.E, & Wang, A. L. (2002). COTS product characterization. *The 14th International Conference on Software Engineering and Knowledge Engineering, SEKE '02, Ischia, Italy, 335*338.

Tripp, L.L. (2002). Software certification debate: Benefits of certification. *The IEEE Software,* 35(6),31- 33.

Van Zeist, R.H.J. & Hendriks, P.R.H. (1996). Specifying software quality with the extended ISO model. *Software Quality Journal,* 5, 273-284.

Vermesan, A. (1998). Software certification for industry - Verification and validation issues in expert systems. *The 9th International Workshop on Database and Expert Systems Application, Vienna Austria.*

Voas, J. (1998). The software quality certification triangle. *Crosstalk: The Journal of Defense Software Engineering*, 11(11), 12-14.

Voas, J. (1999a). Certifying software for high assurance environments. *The IEEE Software,* 16(4),48-54.

Voas, J. (1999b). User participation-based software certification. *Proceedings of Eurovav* 1999, Oslo, Norway, 267-276.

Voas, J. (1999c). Certification: Reducing the hidden cost of poor quality. *The IEEE Software*, 16(4), 22-25

Voas, J. (2000). Limited software warranties: Engineering of computer based systems. *(ECBS2000) Proceedings Seventh IEEE International Conference and Workshop, 56-61.*

Voas, J. (2004). Software's secret sauce: The "-ilities". *The IEEE Computer*, 2(16), 14-15.

Whittaker, J.A. & Voas, J. (2002). 50 Years of software: Key principles for quality. *The IEEE IT Pro*, 4(6), 28-35.

Yamada, A. (1996). *Information technology: Software quality characteristics and metrics part* 2: *External Metrics.* Draft Technical Report ISOIIEC JTClISC7.