User-Centred Software Product Certification: Theory and Practices

Aziz Deraman
University of Malaysia Terengganu, K. Terengganu, Terengganu, Malaysia

Jamaiah Yahaya, Fauziah Baharom
University of Northern Malaysia (UUM), Sintok, Kedah, Malaysia

Abd. Razak Hamdan
National University of Malaysia (UKM), Bangi, Selangor, Malaysia

Abstract

Previous researches on software certification provide models and frameworks that work on traditional approaches such as block box testing, developer's assessment and internal audit of the technical aspects of software. For the past seven years, our research focused on development of software certification product towards user-centred approach. We believe that the technical aspects of quality will be taken care by the technology. Two preliminary works were undertaken to study the issue of certification and as a result, two fundamental models of software certification were constructed successfully. The models have been developed using requirement-design-implementation strategy to ensure that it meets the needs of a number of different interest groups in the industry. These models focused on certifying software by development process and product quality approach respectively. The models have been tested in case studies, which were launched collaboratively with industries in Malaysia. A significant advantage of our model is that it allows users to choose and design their implementation model of certification which fit with their organization's requirements and expectations. This paper presents the opportunity provided by this research that the certification model developed does not only offering a mechanism for certification process, but also providing an alternative mechanism for monitoring of quality and continuous improvement of software quality throughout its life span.

1. Introduction

The failure of many large software projects in term of not meeting user/business requirements, prone to errors etc has led to software quality becoming one of the key issues from all stakeholders' perspectives. As society now becomes dependable on software applications in every aspects of life, questions of quality are becoming important. Some pertaining questions are still being debated such as: (i) determining the quality of software product; (ii) defining mechanism for assessing software product quality; (iii) ensuring and offering software quality guarantee; and (iv) ensuring the continuous improvement of quality of software products. A practical mechanism for assessment is required to resolve these uncertainties. Software quality techniques have been studied using mostly experimental methods manipulating variations within each technique. However, little is known about the practice of quality technique in real context.

Many researchers have acknowledged quality models in literature, quality still remains a complex concept. Quality is the eye of the beholder and it means different things to different people and highly context dependent [1]. Therefore, "Software quality is nothing more than a recipe. Some like it hot, sweet, salty or greasy" [2]. Thus, there can be no single simple measure of software quality acceptable to everyone**.** In addition, we believe that different stakeholders may have diverse interest and objectives through the different roles that the software meant to the organizations. Therefore, understanding the diverse perspectives and expectations represented by the stakeholders will present a more holistic picture of software product quality improvement throughout its lifespan. These works will assist stakeholders to evaluate and assess the quality status of the software operating in their environment at any time. These reasons motivate the study reported in this paper. It presents the quality and certification model and how the certification approach resolves issues that may be caused uncertainties in quality of software during its lifespan. This paper presents too the theory and the practical implementation of software certification process and practices in real environment.

2. The Software Quality Model

Researchers have debated whether quality is defined by the number of defects, quality-carrying properties or quality factors. Several quality factors that can be found from literature are the well-known software characteristics such functionality, maintainability, reliability, usability, reusability, portability and etc. Some researchers focus on the quality of a product together with the process quality. While some researcher focus more on the end

product quality approach which means that relates to quality in-use in certain environment.

As observed from existing quality models for software product assessment, available identified quality attributes is difficult to meet current requirement and specification. Current and available quality models are much dependent on the usage of the assessment process and development requirement. The earliest models of quality such as McCall, Boehm, FURPS and ISO 9126 are limited to measure of external software characteristics such as reliability, maintainability, portability and functionality which do not consider other necessities needs such as conformance of user requirements and expectation. Software quality was more on customer satisfaction and software correctness was not sufficient to be declaring as good quality without satisfaction by the users [3]. This means that there is a requirement to include measurements of human aspects and the quality impact in the new software quality model. Integrity as one of the vital attribute in current situation is not considered in previous models.

SQuaRE is the next generation of ISO9126 model which was built to improve the current needs of software quality requirement. The general objective of this second generation of quality model is to respond to the evolving needs of users through an improved and unified set of normative documents covering quality process: requirements specification, measurement and evaluation [13].

Our research group has developed an integrated and practical model of software quality which is named as Pragmatic Quality Factors (PQF). PQF is the identified factor for quality assessment used in the certification process. PQF consists of two main components: the behavioural attributes and the impact attributes. The behavioural attributes deals with assessing software product to ensure the quality of the software and how it behaves in the environment. It includes efficiency, functionality, maintainability, portability, reliability, security and usability. While the impact attributes deal with how the software react and impact to the environment. These attributes include user perception and user requirement. These two components of quality produce a balance model between technical requirement and human factor.

The attributes defined in this model are considered as the highest level in the hierarchy. These attributes then are broken down into several metrics and measures. The hierarchy model is adopted from software quality metrics framework introduced by IEEE [8]. The measures are the measurable quality aspects in this model. The measurements used are based on perception scales obtained by the assessment team.

Another interesting feature of this model is the weighted scoring method (WSM), which applies different levels and categories of attribute with different weight factors. Literature suggests that each software quality attribute must not have the same level of importance in the real world environment to represent the actual business requirements. Previous study conducted in this research indicated that there were some degrees of considerations or degree of importance of each quality attributes because they were not equally importance between each of them. Later investigation found out that software quality attributes could be classified into three layers of importance namely low, moderate and high [4]. For each layer, a range of weight factor is assigned and recommended. The beauty of this approach is that software owner or the stakeholder has a flexibility and authority to choose relevant weight values to reflect the organisation's and business's requirements and constraints. It is normal in business environment that in some situation certain quality attributes are more important than the others.

3. Theoretical Development

The software quality model, PQF which has been developed in this research is applied in certification process framework. The framework describes the certification process in two different approaches which by means of process and product [6]. Software certification can be viewed in three perspectives which are by product, process and personnel. It is also known as software certification triangle [11]. Software certification can be implemented separately but combination of these approaches produces a more balance result.

Figure 1 encapsulates the high level architecture of our software certification model. That is, our obligations are fourfold:

- to identify software product components. This includes the functional and non-functional characteristics of software product, documentations, data and users.
- to construct the software quality model i.e. the PQF model discussed in the previous section.
- to construct the certification method and procedures.
- to link the software component, model and method to construct the model for software product certification.

Once the framework is constructed that defines and support our software certification model, the final obligation is to define a strategy or approach for implementing this model. The strategy and approach will be discussed in section 4 of this paper.

SCM-*prod* is the model that has been developed by our research team which focuses on certifying software by end product quality approach. In this model, we focus on certifying and assessing software mainly through end users approach. This approach is

adopted because we believe that the technical aspects of software will be taken care by the technologies through the hardware and software standard. It consists of four main components: PQF, product certification repository, certification representation method, and assessment team. The model provides algorithms to measure software quality and software certification level based on identified standard. First algorithm is to measure quality status of each attribute based on the average score in assessment exercise. Second algorithm is to measure the certification level of the software product. Results from both algorithms are mapped into a certification representation model to determine the certification level (1, 2, 3, or 4) and its representation either of 4=excellent, 3=good, 2=basic and acceptable, or 1=poor.

The SCM-*prod* model provides procedures and guidelines for certifying software product operational in certain environment. Interviewee is defined in this model, which identify responsible person to evaluate items in the metrics. Thus, it gives fairer evaluation of the products because it names interviewee based on appropriateness and suitability of the person. The framework is described in detail in separate documents [5],[6].

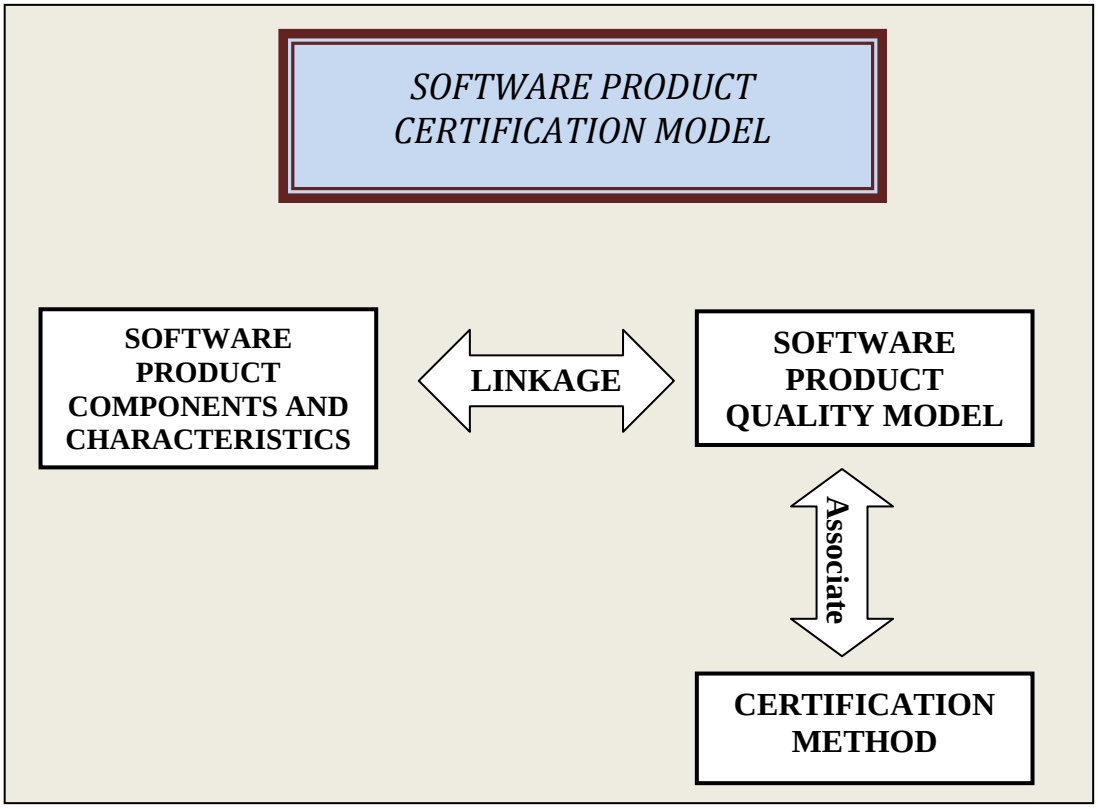

Figure 1. High Level Architecture of Software Product Certification Model

4. The Practices

SCM-*prod* model has been implemented and supported by a toolset which was developed by our research team. The toolset, SoCfes (Software Certifier System) was built using a high level programming language with database system embedded and integrated. See [12] for more detail of the system. The toolset is designed to support flexibility in the implementation model or working model of the certification process. The working model is unique for each organization and therefore, offers flexibility in meeting organization needs and requirements. This flexibility is defined as the capability to select associated and relevance software quality attributes together with their weight values. Furthermore, the toolset was designed to enable users to access their system at their own convenient time.

The implementation and practical application of this model was conducted through a case study which was conducted at a government hospital in Kuala Lumpur, Malaysia (to be referred to as Hospital XYZ). Two years ago in 2007 we have

conducted the first study with Hospital XYZ to assess software that operating in their environment (to be referred to as system *xyz*). During the first assessment time, system *xyz* was just completed in the development and implementation phases. The installation was taken place in the actual environment for about a month from the time of assessment. System *xyz* is a hospital information system which was developed collaboratively with domain expert users from various departments in the hospital. Hospital XYZ is a teaching hospital which provides secondary and tertiary treatment. It is a reference centre for whole country. The hospital provides up-to-date and advance medical services and capable to provide excellent medical services. System *xyz* is a web based application and users can assess the system at any location in the hospital. Modules covered in this phase includes:-

1. Registration & ED Module: For patient registration.
2. Appointment module: For handling information on patient's appointment and accommodating appointment slots.
3. Admission, Transfer and Discharge (ADT) module. This module is used to handle information on bed ordering, ward admission and transfer.
4. OT Scheduling Module: This module functions to manage operation theatres reservation by doctors.
5. Medical Record Module: It functions to handle and manage patient file movement in the hospital.

There were several objectives to be achieved in this exercise. First objective was to assess *xyz* according to SCM-*prod* model of certification. Second objective was to determine the relative certification level in conformity with SCM-*prod* model and standard. The third objective was to evaluate SCM-*prod* model in actual environment. These three objectives were achieved successfully in the first assessment done in 2007 (see [7]).

Assessment study of software product *xyz* was conducted again and repeated in September 2009. The objective of the second study was to determine the current status of quality and certification of system *xyz*. The second objective was to monitor the quality performance of this software after been used for more than two years in actual environment.

4.1. The Approach

The assessment was conducted in three main phases as follows:-

Phase 1: The assessment task started with a meeting with the staff of Information Technology Centre of the Hospital XYZ. The main discussion was about the plan of activity and target respondents. The meeting has concluded and agreed to involve the ENT Clinic and Ophthalmology Clinic as the respondents of this assessment. The committee in the meeting also agreed with the schedule of the second assessment study. Table 1 shows the general activities involved in this phase.

Phase 2: The actual assessment was conducted as schedule. A briefing with representatives of both clinics was implemented to discuss the general procedures of assessment and to obtain detail respondents in the clinics. The respondents were as follows:-

1. Ophthalmology Clinic, Reception Counter
2. Ophthalmology Clinic, Ward Surgery 5
3. Ophthalmology Clinic, Doctors
4. ENT Clinic, Reception Counter
5. ENT Clinic, Ward Surgery 7
6. ENT Clinic, Doctors

Table 1. Activities in phase 1

Activity	Description
Identify organisation and candidate software	To understand organisation's aim and objective. Identify candidate software : • Version • The level and phase of operation.
Design the implementation plan and model	• Create implementation strategy and plan. Brief instruction and meetings to be held. • Identify attributes and weight
Develop assessment team	To identify assessment team members. This includes staff from ICT department, users and independent assessor.
Prepare assessment materials.	To prepare assessment and evaluation forms to be used in the assessment procedure.

At this stage, data was collected through interview, discussion and observation. The detail activities are shown in Table 2.

Table 2. Activities in phase 2

Activity	Description
Identify users	To identify potential users to become respondents in this study. Users are identified according to modules and their expertise in using the software.
Conduct interview and discussion with identified users.	Conduct 'face-to-face' interviews with users. The interviews conducted may also include groups of users from same department and evaluation module. The results attained must be agreed by all members in the discussion group.

Test each module on-line.	Test each module on-line if necessary in attendance of users to protect privacy of data.
Document checking	Data is also collected by document checking. Two main documents are user manuals and design specification.
Data collection	Complete the assessment and data collection
Data checking and validation	Check and validate data gathered in the assessment.

Phase 3: The assessment analysis was supported by the toolset, SoCfeS. Using the toolset, analysis can be done faster, easier, more systematic and accurate. The detail activities are shown in Table 3.

Table 3. Activities in phase 3

Activity	Description
Determine quality score	To determine quality level of each attributes and sub attributes. Total score obtained of each attributes and sub-attributes determine quality levels. To calculate the quality score of software as a product.
Report preparation and presentation.	Graphical representation reports are prepared as follow :- • Radar chart to demonstrate graphically the quality results. • Line chart to demonstrate attributes and their scores obtained in the assessment. • Assessment and certification report.
Determine certification level of candidate software.	Certification levels are defined as follows: - Level 1 = "*excellent*", Level 2 = "*good*", Level 3= "*basic and acceptable*" and Level 4 = "*poor*".
Post-mortem	The committee is encouraged to study the detail of the reports and provide feedback and recommendation to improve the software.

4.2. Findings

This section discusses the result from the analysis. The scores obtained by attribute and sub attributes associated with software *xyz* are shown in Table 4. In this table, there are two scores which refer to previous scores and current scores. The previous scores refer to the score obtained in the study in 2007 and the current scores refer to the study done in 2009. The scores obtained in most of the attributes in current study decrease compares to previous study except maintainability, usability and user factor.

The total quality score as one product is computed and the result is shown in Table 5. The computed score is **74.5** which is equivalent to **Basic and Acceptable**. In this analysis we use the same weight value for each behavioural factor in both assessment studies. The analysis shows that there is not much difference in the quality score resulted in both studies.

Figure 2 illustrates the quality attribute's scores in more detail in Kiviat charts. In Kiviat chart the points fall in outer layer are considered better than the points in the inner layers. Each attribute is represented by axis and scores are plotted at the limits between 0-100%. Kiviat graph can be used to easily identify attributes that need attention in this process. Attribute that fall on the limit's outer layer is considered better quality compares to attributes at inner layers of this graph.

The chart in Figure 2 demonstrates the results in both studies (year 2007 and 2009). The assessment study in year 2007 is labeled in dotted line while the

Table 4. Results in previous and current study

	Previous (2007)		Current (2009)	
Quality Attribute	Score /5.00	%	Score /5.00	%
Efficiency	4.08	81.6	3.05	61.0
Time behaviour	*4.33*		*3.25*	
Resource utilization	*3.7*		*2.75*	
Functionality	3.69	73.8	3.33	66.7
Suitability	*3.65*		*3.41*	
Accuracy	*3.20*		*3.38*	
Interoperability	*4.50*		*3.13*	
Maintainability	2.66	53.2	3.35	66.9
Analysability	*2.63*		*3.43*	
Changeability	*2.20*		*3.21*	
Testability	*3.06*		*3.33*	
Portability	3.55	71.0	3.47	69.4
Adaptability	*5.00*		*3.67*	
Installability	*1.80*		*3.09*	
Conformance	*4.80*		*3.50*	
Replaceability	*4.40*		*4.00*	
Reliability	3.36	67.2	3.14	62.8
Maturity	*3.80*		*3.44*	
Fault Tolerance	*3.20*		*3.03*	
Recoverability	*3.00*		*2.89*	
Integrity	3.83	76.6	3.08	61.7
Security	*3.87*		*3.17*	
Data Protection	*3.06*		*3.00*	
Usability	2.95	59.0	3.71	74.3
Understandability	*3.44*		*3.78*	
Learnability	*2.93*		*3.79*	
Operability	*3.01*		*3.63*	
User Factor	3.67	73.4	4.18	83.5
User's Perception	*3.84*		*4.25*	
User Requirement	*3.40*		*4.06*	

study in year 2009 is labeled in a straight line. It is clearly shown in the chart that most attributes (portability, reliability, efficiency, integrity and functionality) fall in the inner layer compare to the previous assessment. This shows that the quality performance of the attributes is degraded from years 2007 to 2009. Whilst the other attributes: maintainability, usability and user factors have improved.

Table 5. Results in previous and current study

Behavioural Factors	Max Value (1)	Weight (2)	**Previous (2007)** Score Obtained (3)	Score (4)	Quality Score (%) (5)	**Current (2009)** Score Obtained (6)	Score (7)	Quality Score (%) (8)
Efficiency	5	7	4.08	0.539	10.8	3.05	0.403	8.1
Functionality	5	9	3.69	0.627	12.5	3.33	0.566	11.3
Maintainability	5	7	2.66	0.351	7.0	3.35	0.442	8.8
Portability	5	4	3.55	0.268	5.4	3.47	0.262	5.2
Reliability	5	9	3.36	0.571	11.4	3.14	0.533	10.7
Usability	5	7	2.95	0.390	7.8	3.71	0.490	9.8
Integrity	5	10	3.83	0.723	14.5	3.08	0.582	11.6
TOTAL		53		3.468	69.4		3.278	65.6
Impact Factors								
User Factors					73.4			83.5
Total Product					**71.3**			**74.5**

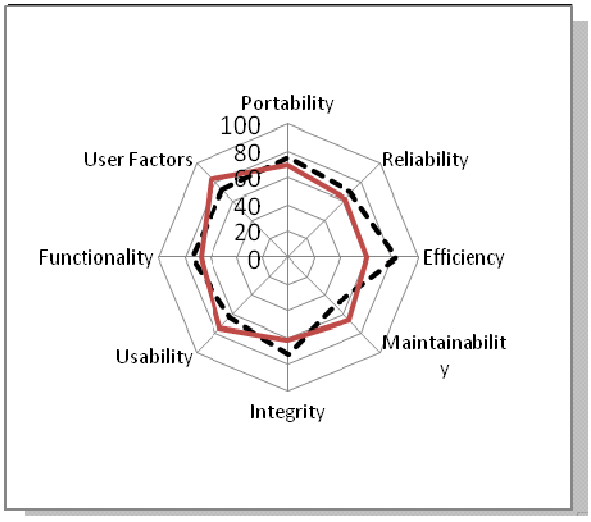

Figure 2. Kiviat chart of system *xyz* quality attributes

4.3 The User-Centred Approach of Software Certification

The score of user factor increases from 73.4 (previous study) to 83.5 (current study) which demonstrates the acceptable satisfaction in user perception and requirements. Even though there are some weaknesses and problems of the system, which have been identified by this study but the users still believe that the system is beneficial and useful to support their everyday tasks. This improvement influences the overall total quality score of the product. The certification level maintains the same achievement as in the previous assessment, **basic and acceptable**. On the other hand, the result of the behavioural attributes which refers to the software quality in-use is degraded from 69.4 to 65.6.

The problems that being identified are as follows:-

- System hangs and downs for some period of time. The recovery time normally long and it ranges from several minutes to one or two days. This problem causes the low score in efficiency and reliability. It is suggested that a client charter needs to be introduced to improve trustworthiness of this system to the users in Hospital XYZ.
- The system's operation normally is not as good as the user expected especially in the morning of the day. The transactions operate very slow and poor during this period of time.
- Problem with delay in creation new username and password especially for doctors. Many situations doctors need to use and share username with senior doctors to enter and use the system. This cause implication of security, integrity and also accuracy of the system.

- Faulty in transactions, i.e. transaction bills are printed double and patient registration numbers are duplicated. For both situations, a human decision is required to decide which registration number is the valid number and second, the user needs to delete which row that is not necessary to be included in the bill. Therefore, the functionality and reliability of the system are degraded.
- Functionality aspect such as option for others is not available, alert system for OT monitoring system and issue in closing schedule of OT system. These problems are highlighted by the doctors.
- Integration with other systems operating in Hospital XYZ. This cause difficulty to the users to switch from one system to another. It is suggested that we have a one-login system for all systems in Hospital XYZ.
- There is no facility that support and benefit the doctors for their research activity to maximise use of *xyz* data. The function like statistical report of the *xyz* data may be useful for the doctors. This relates to the user factor of quality (user perception and user requirement aspects) in the assessment.

In this approach users of the system that being certified are the main contributions to the assessment. This to ensure that the system meets the user expectation and requirement and can be justified and confirmed through the certification exercise.

5. Discussion

As Chinese proverb says that there is always a first step in a journey of ten thousand miles. Thus, during the last seven years, our team started to focus and concentrate our research on development of the framework of software product certification. Two fundamental models was constructed which by means of process and product quality approach. These two models (SPAC [10] and SCM-*prod*) were successfully developed, tested and applied in real environment via industrial involvement. The collaboration involves several large organisations in Malaysia to practice using the models in the real environment where the systems are operating. The model is implemented in a collaborative perspective approach which includes the users, developers and external assessor (independent assessor) but focused more toward user-centred approach.

The theory and methods discussed in this paper should be generalized and suitable for any organizations and systems operating in the environment. It has been tested in the business environment successfully. The experience gained through these practical applications have been analyzed and recorded into a formal certification database and documentation. The results reported so far show significant satisfaction of software developers, managers, and stakeholders, who feel more confidence in using this model for assessing and certifying software product. With these model, specifically, SCM-*prod* model that is discussed in this paper, the quality status of the product can be identified and justified. The assessment and certification can be repeated several times during the life span of the products.

Our certification experiences have been shared with several software practitioners in selected organizations in Malaysia through transfer technology workshop that was conducted recently. This strategy of transferring the technology will be conducted again in the near future through a series of certification workshops.

6. Future Work

Our next effort in this research is to develop a software-aging index through certification history. Software products do exhibits a behaviour that closely resembles human aging. Like people, software gets old and we can't prevent aging, but we can understand its causes, take steps to limit its effects and prepare for the day when the software is no longer viable. There are two distinct types of software aging. The first is caused by the failure of the product's owner to modify it to meet changing needs; the second is the result of the changes that are made [9]. This may lead to rapid decline in the value of a software product and, thus accelerating the aging process. Few major causes in software aging are dealing with the quality of documentation, design for change in software code and unimposing standard.

Previous study indicated that software practitioners were not keeping good and up-to-date records in documentation and were not giving serious attention on imposing standard in software development [10]. Therefore, in this situation, certification can be used to enforce continuous quality improvement through periodic certifying process. Certification level and status at certain time (t), Ct can be obtained and comparable at another time interval ($t+i$), C$t+1$. Collection of certifying information could be used as the input for designing a software-aging index. Thus, estimation of its aging could be achieved through this index and we could limit its aging effects and decay, and perhaps delay the aging of the software.

Secondly, we will extend this model and implementation in a different domain and scope. The next strategy is to enhance the model and practice in e-government applications. In doing this, several objectives have been identified to answer some research questions such as how does the quality of an e-government applications perceived by the users

and managers, and how does these requirements suit with the fundamental model of software certification.

7. Conclusion

A framework and model that may be used to certify software based on product quality approach has been presented in this paper. The model, SCM-*prod* is a practical model of certification, which has been evaluated and tested, in real case studies involving three large organizations in Malaysia. These practical applications were conducted in 2007. We have repeated the software assessment and certification in one of these organizations in year 2009. The results showed that after two years there was not much different from the first result as explained in section 4. The industrial applications determine that the model is practical and can be applied in the real practice. The significant advantage of this model is that it can be tailored and flexible to the organisation's requirements. Therefore, the working model can be developed based on organisation's demands and constraints. Second advantage of this model is the involvement of users in the certification process that justify the confirmation of user expectation and requirement for software quality. With the involvement, the certification process is moving towards user-centred approach.

8. Acknowledgement

This research was funded by The Ministry of Science, Technology and Innovation of Malaysia. Our thanks go to the Universiti Kebangsaan Malaysia Medical Centre who provided the necessary insight into the case study.

9. References

[1] B. Kitchenham, and S.L. Pfleeger, "Software quality: The elusive target", *IEEE Software*, January, 1996, pp. 12-21.

[2] J. Voas, "Software's secret sauce: The "-ilities". *IEEE Computer* (November/December),2004, pp. 14-15.

[3] P.J. Denning. "What is software quality?", *A Commentary from Communications of ACM,* January, 1992.

[4] J.H. Yahaya, A. Deraman, and A.R. Hamdan, "Software product certification model: Classification of quality attributes", *The First Regional Conference of Computational Science and Technology (RCCST 07), Kota Kinabalu,* 2007, pp. 436-440.

[5] J.H. Yahaya, A. Deraman, and A.R. Hamdan, "Software Certification Model Based on Product Quality Approach", *Journal of Sustainability Science and Management,* 3(2), December 2008, pp. 14-29.

[6] J.H. Yahaya, A.Deraman, F. Baharom, and A.R. Hamdan. "Software Certification from Process and Product Perspectives", IJCSNS International *Journal of Computer Science and Network Security,* Vol. 9 No. 3, March 30 (2009). ISSN: 1738-7906.

[7] J.H. Yahaya, A. Deraman, and A.R. Hamdan, "Software Certification Implementation: Case Studies Analysis and Findings", *The 3rd International Symposium on Information Technology 2008 (ITSim2008),* 26-29 August, 2008, Kuala Lumpur, Vol. III, pp. 1541-1548.

[8] IEEE: IEEE standard for a software quality Metrics Methodology, http://ieeexplore.ieee.org/xpl/standards.jsp, 1993. Access date: 20 August 2005.

[9] D.L. Parnas, "Software Aging." *IEEE,* 1994.

[10] F. Baharom, A. Deraman, and A.R. Hamdan. "A Survey on the current practices of software development process in Malaysia." *Journal of Information and Communication Technology (Journal of ICT)* Volume 4, 2005, pp. 57-76 (ISSN :1675-7505).

[11] J. Voas, "The Software Quality Certification Triangle". CrossTalk, *The Journal of Defense Software Engineering*, November 12-14, 1998.

[12] A. Deraman and J. Yahaya, "The Architecture of an Integrated Support Tool for Software Product Certification Process" *Journal of Information & Systems Management* Volume 1, Number 1, March 2011, pp. 46-56.

[13] W. Suryn and A. Abran. "ISO/IEC SQuaRE. The Second Generation of Standard for Software Product Quality", *IASTED 2003-SEA 2003*, November 3-5, 2003 Marina del Rey, CA, USA.

