Software Certification from Process and Product Perspectives

Jamaiah Haji Yahaya†, Aziz Deraman†† , Fauziah Baharom† and Abdul Razak Hamdan††††

†College of Arts and Sciences,
Information Technology Building, Universiti Utara Malaysia (UUM), Sintok, 06010 Kedah, MALAYSIA
††Academic and Internationalisation Division,
University Malaysia Terengganu, Kuala Terengganu (UMT), 21030, Terengganu, MALAYSIA
†††Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM), Bangi,
43650 Selangor, MALAYSIA

Summary
Previous studies by other researchers in this discipline provide a set of axiom and supporting models for software assessment and quality but are not extended to a practical and well accepted model of certification. During the last five years, two preliminary works was undertaken in this research group to study issues of certification and these works are still continuing. Recently, this initial work was completed and extended. Two software certification models have been developed using requirements-design-implementation strategy to ensure that it meets the needs of a number of different interest groups in the industry. The two models focused on certifying software by development process and product quality approaches. The models have been tested by case study, which was launched collaboratively with industry in Malaysia. Further analysis has demonstrated feasibility and practicality of the models in a real environment. This paper discusses the implementation of these two models (SCfM_prod & SPAC) and their underlying results. The certification models defined in this research do not only offering a mechanism for assessment and certification, but also providing an alternative mechanism for monitoring of quality and continuous improvement of software quality throughout its life span.

Key words:
Software certification model, collaborative perspective, process development approach, product quality approach, software assessment, software certification

1. Introduction

Chinese proverb says “there is always a first step in a journey of a ten thousand miles”. In research we always start with an initial work in specific area or study. The initial works are usually far from the targeted outcomes and may take several years to complete. In software certification, theories and axioms are gathered and constructed before the implementation put into practices. A few models have been introduced in literature with limited and unknown success. Some suggestion reasons for this are:-

- The proposed models have not been underpinned by a sort of empirical theory and industrial observations.
- There are number of different aspects of quality properties that are known to be positively influence its quality but these properties have never been organized into a sort of systematic framework.

Our claim is that these matters are properly attended to it is possible to construct a practical model of software certification. We employ a goal-directed requirements-design-implementation strategy to develop a model for software assessment and certification that will attend to these matters.

The broad ideas of software product certification were narrowed down into two main studies. These studies were conducted underpinning the requirement that “good quality process of development will produce good quality software product”. Our obligations are fourfold:

- To identify software certification requirements from the environment and industry
- To construct a software certification model based on process development quality approach
- To construct a software certification model based on product quality approach
- To validate and evaluate the models through case studies involving organizations and industries in Malaysia.

The first task in building a software product certification model is to identify the requirements through empirical studies. Two background studies via surveys were conducted to investigate and identify the requirements of best quality software and certification in the industry [1,2]

Manuscript received March 5, 2009
Manuscript revised March 20, 2009

2. Issues in Software Certification

Quality and certification are two tightly coupled terms, which need to be considered one in another. General expressions of how quality is realized in software are with "fitness for use" and "conformance to requirements". "Fitness for use" refers to characteristics such as usability, maintainability and portability. "Conformance to requirements" means that software has value to users [3]. International Organisation for Standardisation (ISO) defines quality as "the totality of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs" [4]. IEEE defines software quality as – a software feature or characteristic used to assess the quality of a system or component [5]. Quality is in the eye of the beholder because different people may view quality in their own ways or perspectives.

Software certification is a new concept in Malaysia but increasingly popular in Europe and United States. Many debates on this issues are reported but at the same time communities are starting to accept this concept in software industry (see also [1, 2]). Software certification is a written assurance that a product, or services conforms to specified characteristics. Assurance and conformation are normally provided by a third party organization. Software certification is the extended of quality by means that quality need to be measured prior to certification granting process. Software certification can be viewed in three different perspectives: personnel, process and product and also known as certification triangle [6]. The combination of these three will produce a best balance result [7, 8]. Having a certification environment in software industry, organizations will give more emphasize on standard in their processes and procedures [26], encourage continuous improvement [29] and improve user confident toward the quality of the software [27].

One possible approach in implementing certification is through involvement of end users in the process by delivering information regarding the usage of the software [8], developers self certification [9] and verification and validation technique [10]. The mentioned researches are associated with certification of software products.

3. Software Certification (SC) Framework

A framework of software certification environment has been designed and showed in figure 1. The framework shows a combined or joint certification criterion to produce a comprehensive certification of candidate software product. The certification level of software product is obtained by means of two distinct approaches of certification, process and product approach.

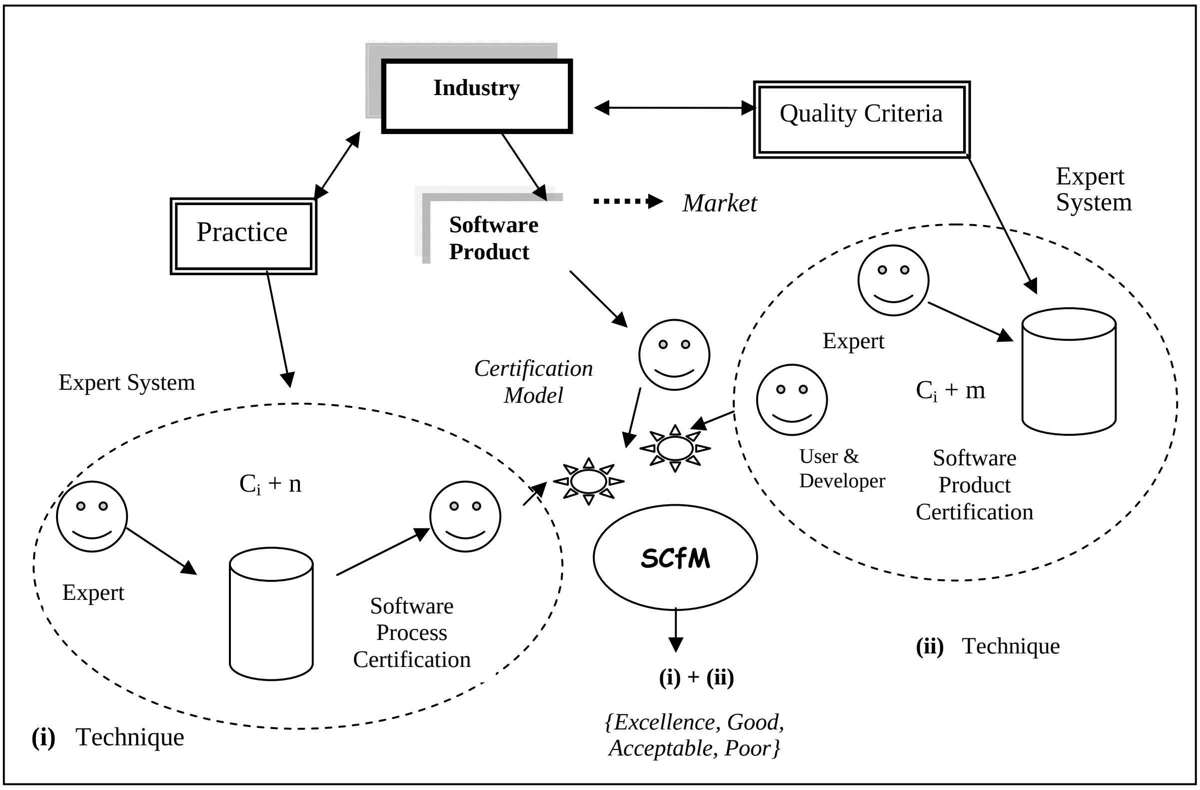

Figure1: Software Certification (SC) Framework

Despite several methods to assess software, we investigate the possibility of conducting assessment and certification of software product using collaborative perspective approach, which consists of users, developers and independent assessor. The advantages of this approach compares to other approaches are: - 1) this approach eliminates bias assessment and evaluation of the product by including independent assessor in the team, 2) removes unfairness evaluation by including the owner or users of the product to participate in the assessment process and 3) accelerates the process because the team is familiar with the product and its' environment.

Two main models have been developed and constructed associated with two distinct approaches defined in the framework. The two models are named as SPAC and SCfM-*Prod* which are certification by development process and end product quality approach respectively. Each model has separate quality criteria and performs discrete procedures and tasks in the assessment and certification process. A certification expert system defined in this research is an automated and intelligent tool to support, accelerate and manage the process based on current and future requirements. The existence of the expert system will improve the environment and also improve the involvement of participants of various people in the certification process.

The framework is appropriate to assess and certify completed software product. This framework is beneficial to the end users as well as publishers, software's owner and stakeholders. This persuasive feature of this model is obtained through involvement of independent assessor, users and software developers in the assessment team. These models will be discussed in detail in the next sections.

3.1 Software Process Certification Model (SPAC)

The first certification model developed in this research is SPAC Model – *Software Process Assessment and Certifcation Model.* The primary goal of this model is in assuring that the software development process are carried out effectively and efficiently to meet the expected quality criteria, delivered on time and within budget. This model is formulated based on the existing models, which are Capability Maturity Model (CMM) [11], ISO 9000, ISO/IEC 15504 [12] and Bootstrap [13]. It is mainly focused on five key factors that influenced on the quality of software. The factors are the quality of process performed the quality of people involved, the use of development technology, the stability of working environment and project conditions.

SPAC consists of seven components, which can elaborated as the candidate software, the process quality factor, the certification and quality index, assessment team and repository. This model is demonstrated as in Figure 2.

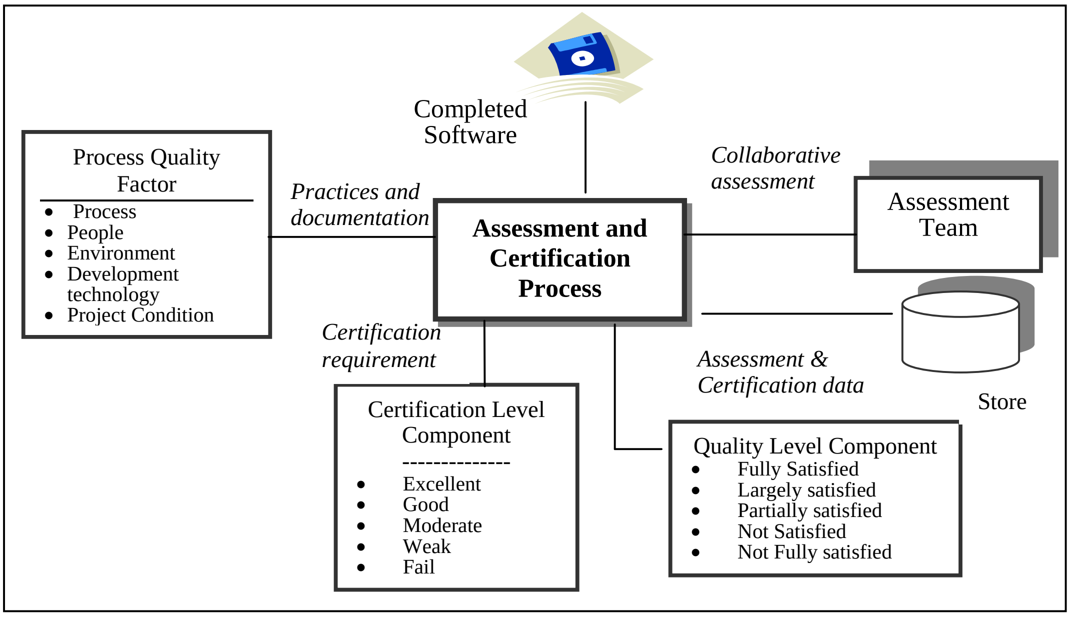

Figure 2: SPAC model [14]

a) The first component of SPAC is the Process Quality Factor (PSQF). It defines what to be measured in this model. PSQF identifies factors that affect the quality of software process in practice. The five factors are: process, people, environment, development technology and project constraint.

- Process: The factor of process includes three basis activities, which are development, management and support activities.
- People: This factor measures in term of skill, experience, knowledge, team commitment, user involvement and management responsibility.
- Environment: This factor measures the comfort ability and safety aspects in the work place.
- Development technology: This factor measures in term of standard and procedure, tools, methods and techniques and process origin.
- Project constraint: This aspect of quality measures the time delivery and budget.
-

b) The second component is the candidate software to be assessed. This candidate is a completed product that is ready to be delivered to users or customers. Information on the development process is collected via multiple techniques: reviewing all artifacts produced during development process, interviewing key personnel and also observing the working environment.

c) The assessment team is the third component of this model. In this model assessment is carried out in a group by a collaborative approach. Developers, independent assessor and project manager should be apart of the assessment team and the team's leader must be an expert in software engineering and software quality.

d) The forth component is the assessment and certification process. This component contains three main phases of implementation: preparation, execution and post assessment phase. These phases are then decomposed into 16 activities that provide guidance to facilitate the whole process of certification.

e) The fifth and sixth components are the quality and certification level.

f) The seventh component is the repository, which stores all information and results from assessment and certification exercises. This data is useful for future analysis and improvement.

As mentioned earlier in this paper, SPAC is implemented in collaborative assessment approach. The assessment is conducted in the perspective of independent assessor and the development team. This model introduces two main quality indexes, which are quality assessment level and certification level. Quality assessment level is used to identify the achievement for each attributes and metrics. The achievements obtained in the assessment exercise are useful for the organization to plan for future improvement. The candidate software developer group uses certification level to picture the overall performance of software process development practices. This model adopts the CGPA approach to identifying and naming of the certification levels.

This model has undergone an evaluation through case studies conducted in Malaysia. This will be discussed in section 4.

3.2 Software Product Certification Model (SCfM_*Prod*)

The second model of certification developed in this research is named SCfM_*prod*, which focuses with the certification model by product quality perspective. The certification by product quality approach is an acceptable alternative of certifying software and the underlying hypothesis is that good software development processes do not guarantee the excellent quality of product. Therefore, assessment of end product software must be independent from the development process. Previous studies [15,16] show that code analysis and testing software alone will not guarantee the quality of the product. Many defects cannot be found through code analysis because they reflect tacit or undesirable requirements or can be observed only when the product is being used.

SCfM_*prod* model consists of six main components: pragmatic quality factor (PQF) as the quality certification guidelines and standard, product criteria, certification specification, certification representation method, repository and certification team.

The first component of this model is the pragmatic quality factor (PQF) which is the quality certification guideline and standard for measuring software product quality. Undertaking quality attributes defined in ISO 9126 model as the based line of the assessment metrics, we define two sets of attributes, which by means of the behavioural and the impact attributes. The behavioural attributes consist of high level software quality characteristics which include usability, efficiency, functionality, maintainability, portability, integrity and reliability. Integrity is not included in ISO 9126 model but included in this model because of the requirement from literature and empirical study. In the age of hackers and firewalls, the importance of integrity aspect has increased [17] and ISO 9126 model is a generic model but requires some customization for particular case [18]. Integrity measures the ability to with-stand attack on its security that comprises of program, data and document. It covers threat and security aspects. Findings from previous empirical study also indicated the importance of integrity in software quality metrics [1,19]. Each attributes in PQF is made up of several subattributes and then broken down into several metrics that shows the measurement aspects of the attributes.

Survey done by this research group also indicated that quality attributes can be classified into different levels and weights (see also [19]) based on their importance and significance during quality assessment by respondents. Thus, adopting functional point approach, attributes are classified into three distinct classification layers: high, moderate and low. The attributes are grouped and weight factors are assigned as shown in Table 1.

The second measurement in PQF is the impact attribute. This attribute indicates the conformance in user requirements, expectation and perception. These attributes include measure of popularity, performance, trustworthiness, satisfaction and user acceptance. These attributes align with the definition of quality that quality must conform and correspond to requirements and fitness of use. These two groups of attributes are important to balance the assessment between the technical aspects of quality and the human factors [20]. Similar to behavioural attributes, the impact attributes are made up of several subattributes and metrics that show the measurement of the attributes.

Table 1: Classification of attributes and its weight factor

Level	Attributes	Weight Factors
Low	Flexibility Intraoperability Interoperability Portability Survivability	1-4
Medium	Safety Efficiency Maintainability Usability	5-7
High	Functionality Reliability Integrity	8-10

The product criteria component in this model offers services for weight factors as mentioned above and criteria selection. In this model users may select their interested attributes of quality to meet their organizational requirement and target. This offers flexibility in the certification exercise because “software quality is nothing more than a recipe. Some like it hot, sweet, salty or greasy” [28].

The third component is the certification team. As discussed in previous section, the certification process is done collaboratively with three different assessors: the independent assessor, developer and user.

The fourth component is certification specification. This component explains the processes, algorithm, formulas and reporting format in the certification exercise.

The fifth component is the certification representation method. This component offers certification-mapping process to obtain the associated certification level of software product (see Table 2). It is important to note that the ranking of certification level mentioned here is flexible and does not fixed to the stated figures. They are opened for customisation and tailored to requirement by the organisation. The organisation and the owner of the products may decide to modify and customise the classification levels based on their maturity and the readiness of the organisation itself.

The certification levels are identified and characterised in four distinct levels: excellent, good, basic and acceptable, and poor. The certification level of product is determined by comparing the score value (TQP) obtained in the certification exercise. For TQP value greater than 90% and less than 100%, the product obtains a certification level of excellent. This means that the software product satisfies all quality criteria and achieves quality level of excellent and satisfactory. Whilst if the TQP score is greater than or equal to 75% and less than 90, the product is classified as good which means that it satisfies the quality level of good. If the product gains TQP score greater and equal to 50 and less than 75, the product is identified as basic and acceptable which means that the software satisfies the quality level of basic or average and acceptable. Whereas, if the TQP score obtains less than 50, the product is identified as poor and unsatisfactory. The classification level is shown in Table 2. The similar classification technique is used in [21]. The detail algorithms and processes can be found in [22].

4. Software Certification Implementation and Practices

The certification models (SPAC and SCfM-*Prod*) have been evaluated in a case study that was launched collaboratively with a large organization in Malaysia. Result from the study shows that the evaluated aspects in this model are feasible and demonstrates the potentials and practicality of the model in supporting certification methodology. The models facilitate a systematic and repeatable software assessment and certification during its life span. Furthermore, the relationship between attributes, sub-attributes and metrics that measures the quality aspects of software are valuable and provides awareness of improvement in software product quality in future.

Table 2 Ranking of certification levels

TQP Score (*TQP*)	Certification Level	Certification Status	Description
90<= *TQP* <=100	4	Excellent	Software satisfies all quality criteria and achieves quality level of excellent.
75<= *TQP* < 90	3	Good	Software satisfies and achieves the quality level of good.
50< = *TQP* < 75	2	Basic and Acceptable	Software satisfies and achieves the quality level of basic which also means average and acceptable.
0 <= *TQP* < 50	1	Poor	Software attains quality level of poor and unsatisfactory.

4.1 Evaluation Of SPAC Model and Analysis

Case study was conducted in a semi-government organization in Malaysia, which we referred as Case X. Case X is in organization that main functions are deposit and saving services for Muslims, investments and financial services, and pilgrims' services in Malaysia and Mecca. The assessed system is a large system and operating in the headquarters as well as integrated to various branches throughout Malaysia. During assessment and certification exercise, the data was collected through multiple data gathering techniques, which included document reviewing, interviewing and observing. The results of the analysis are explained as follows:-

a) Quality of Process

- Development process – the analysis shows that in this factor, only the requirement management that achieves level of satisfactory. The other attributes, design, coding and testing do not obtain level of satisfactory in the quality assessment level.
- Process management – in this factor, there are five attributes, which consists of project management, change management, quality management, technical management, technical validation and risk management. Each attribute obtains level of satisfactory except that quality management that obtains level of very unsatisfactory level.
- Support process – Support process consists of resource management, training management, staff affair and documentation. All attributes in this factor achieve level of satisfaction.

b) Quality of People

- Software Developer – the quality aspect of the people which involved in development process is measured through several metrics such as group commitment, experience, knowledge, technical expertise, management expertise and interpersonal expertise. In this factor all attributes mentioned above obtain score more than 60% which equivalent to satisfaction.
- User and Management – in this case study, the involvement of management is considered as average and may need more participant and involvement in the future.

c) Quality of Technology

The third factor involved in this assessment and certification is the usage of technology. The analysis shows that the usage of prototyping method and the usage of development methodology are very satisfactory. While in the attribute of tool and technique, the quality level obtained is satisfactory. In term of standard and procedure, it obtains very unsatisfactory level in the assessment.

d) Project Constraint

In this factor, two attributes are measured. First is the scheduling and secondly is the budget. The analysis shows that both attributes obtain level of satisfactory and very satisfactory respectively.

e) Quality of Work Environment

The last factor in this model is work environment. This factor obtains level of satisfactory in this assessment.

Following the assessment and obtaining the quality level of all attributes, the next step is to compute the certification level. The certification score obtained in this exercise is 2.66, which is at level 3 and equivalent to average. The detail steps, algorithms and procedures are explained in [14, 25].

4.2 Evaluation of SCfM_*Prod* Model and Analysis

This section explains the analysis and findings from data analysis of this case study using SCfM_*Prod* model. The same case study, Case X, was used in this exercise. The first analysis is to obtain the score by each attributes defined in this model. The data collection was conducted through a collaborative perspective assessment among members in the team which consists of the independent assessor, developer and users. Following the data collection, data analysis was conducted and the results were shown as in Table 3. The result shows that all attributes achieve scores in range average (between 64.0% - 74.6%). The scores can be mapped into the certification and quality level as been defined in Table 2.

Second analysis is to plot the score in a Kiviat graph. 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. In Case X, efficiency, functionality and integrity fall in better quality level compare to maintainability, usability, portability and reliability.

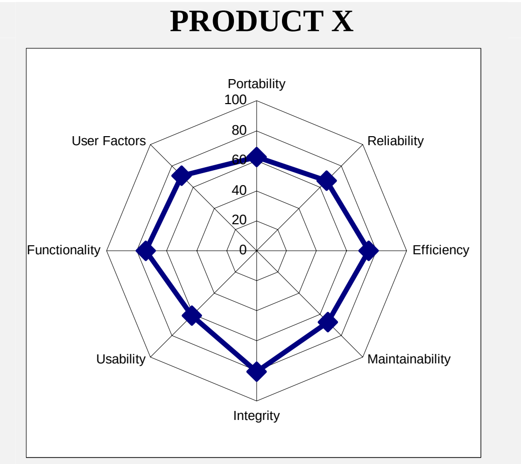

Figure 3: Kiviat Chart of Case X

Table 3: Scores by Quality Attributes

Attribute	Score	Attribute	Score
Efficiency	74.6%	Functionality	72.4%
Maintainability	67.8%	Portability	64.0%
Reliability	66.0%	Integrity	73.4%
Usability	64.0%	User Conformity	70.6%

It is useful to tabulate the results in the previous sections into a summary table for clarity. The summary of all the results is shown in Table 4. Product X of Case X was six months old during the assessment period. This product was developed thru out-sourcing and jointly with another software company. The result shows that product X achieved level 2 of certification with score of 70.08/100, which refers to *basic and acceptable.*

Table 4: Summary of case X

Criteria	
Sector	Business
Software	Human Resource System
Development Approach	Out-Source & Joint Development
Duration of Use	6 months
RESULTS OF ASSESSMENT AND CERTIFICATION	
Quality Score	70.08/100
Certification Level	2
Certification Status	Basic and Acceptable

There are at least two factors to be considered that influenced the certification level of a software product candidate. The two factors are the operation period in the environment of the candidate and second, the weight factors of attributes assigned by the owner of the candidate product. The studies show that the longer the operating period of the software the better result of quality and certification level can be achieved. Clearly, this is true because the software has been updated and corrected accordingly and necessarily by the developers. This relates to the issue of maturity of the software. On the other hand, if the certification exercise is conducted periodically over some time intervals, an unexpected result may be seen because of the aging of the software.

The second aspect that influences the result is the weight factors of the product. Without assigning weights factors to quality attributes, the results may indicate different level of certification. Thus, this model accommodates weight factors for all attributes with different level of importance to reflect individual business requirements [23,24]. It is important that the weight factors are identified and assigned accordingly by the owner of the product to reflect the actual quality status of the software based on the organization requirements and constraints.

5. Discussions and Future Work

The works describe in this paper participated in solving problem in ensuring and determining quality of software product. The product, which was assessed and certified through this case study, has demonstrated some interesting observations. For example, compares to other attributes, this product is weak in maintainability, usability, reliability and portability. This is true because in the assessment by development process approach shows that the product is developed in an unsatisfactory condition. It does not satisfy the requirement and development process standard defined in SPAC model. Maintainability attribute is closely related to design, coding and testing aspects during development. If these activities are not properly developed and implemented, we can expect that the maintainability will be difficult and complex. Thus, the quality in term of maintainability is not good in this case study.

In this assessment, the assessed product is weak in usability aspect. It only achieves 64% based on 100% perception by collaborative perspective assessment. It is true because if we see the assessment through development process approach, this case also obtains average score in user and management contribution in the development. This reveals that the software was developed and implemented without enough involvement of users and management therefore the usability aspect is not considered adequately and sufficiently.

Even though there are several interesting finding and observations in the issues of quality attributes and processes in this case study but further study need to be implemented to verify the correlations between them. This aspect is not covered in this research.

SCfM_*prod* and SPAC model have been developed, tested and evaluated. The implementation of this model can be conducted several times during its life span. Therefore the owners of the software are able to monitor the progress and performance of their software product operating in certain environment. This certification environment supports a continuous improvement during the life span of the product [29]

The certification models discussed in this paper have been developed to work out with current requirements on certification and quality issues. Both issues can be complying with these models but future work needs to be implemented to overcome some of their limitations.

Specifically, one of the limitations is tied with the property of SCfM_*prod* model. The pragmatic quality factor (PQF) applied in SCfM_*prod* model is enhanced from ISO 9126 model with additional features and capabilities. PQF covers both human and technical aspects thus provide better balance in software quality assessment. It is believed that PQF adds value with its human aspect included in the measurement. On the other hand, it is a static model of quality even though it provides some flexibility to the organization in the certification exercise. This quality model is unable to improve its' components or characteristics according to current and future requirements. The model also may not be able to handle multiple assessment and certification exercises easily and efficiently. Therefore, we continue to extend this model and apply in a more comprehensive and integrated model of certification. The future model of certification has an intelligent capability and capable to improve itself in the environment. More documents will be published in the near future on this new design of certification model and intelligent system.

6. Conclusion

A framework that may be used to certify software product has been presented and applied. The framework has been developed in a goal-directed way in order to meet the needs of the different interest groups associated with software quality. This paper discusses our experience in applying software certification in real environment and using two practical models of certification. Thus, with the framework discussed in this paper, certification exercise can be done in two approaches or perspectives: end product and development process. Further analysis and study need to be carried out to investigate the correlation between the quality of the process and quality of the product.

Further more, in this framework we focus on continuous improvement in two facets. Firstly is the continuous improvement of the software product itself. This model is enable easy assessment and certification exercises and offers better guidance and procedures. Having the model and intelligent toolset to support the certification process allow users to evaluate and assess the software continuously, thus facilitate the continuous improvement of the software. Secondly, it provides continuous improvement of the quality model applied in the certification process. The second is still in-progress and need to be tested very soon.

References

[1] Yahaya, J.H., Deraman, A., Hamdan, A.R. "Software Quality and Certification: Perception and Practices in Malaysia" *Journal of ICT (JICT),* 5, Dec, 2006.

[2] Baharom, F. , Deraman, A. and Hamdan, A.R. "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).

[3] Tervonen, I. "Support for Quality-Based Design and Inspection", *IEEE Software,* January, 44-54, 1996.

[4] Yamada, A. (1996). " Information Technology – Software quality characteristics and metrics Part 2 – external Metrics", Draft Technical Report ISO/IEC JTC1/SC7.

[5] IEEE (1993). " IEEE standard for a software quality Metrics Methodology", http://ieeexplore.ieee.org/xpl/standards.jsp. [23 July 2008].
[6] Voas, J. "The Software Quality Certification Triangle." Crosstalk, *The Journal of Defense Software Engineering*, November 1998:12-14, 1998.
[7] Bazzana, G., Andersen, O. & Jokela, T. "ISO 9126 and ISO 9000: Friends or Foes?" *IEEE Software Engineering Standards Symposium*, 1993.
[8] Voas, J. "User participation-based software certification", *Proc. of Eurovav*, June, 267-276, 1999.
[9] Morris, J., Lee, G., Parker, K., Bundell, G.A., & Lam, C.P. (2001). "Software Component Certification", *IEEE Computer*, September, 30-36.
[10] Heck, P & Eekelen, M. v. (2008). "LaQuSo Software Product Certification Model (LSPCM)" *http://alexandria.tue.nl/repository/books/633706.pdf*. [16 July 2008].
[11] Paulk, M. C., Curtis, B., Chrissis, M. B., and Weber, C. V. The Capability Maturity Model for Software, *IEEE Computer Society Press*: 427-438, 1997.
[12] O'Regan, G. *A Practical Approach to Software Quality*, Springer, 2002.
[13] Kuvaja, P. BOOTSTRAP 3.0 - A SPICE Conformant Software Process Assessment Methodology, *Software Quality Journal* **8**: 7-19, 1999.
[14] Baharom, F., Deraman, A. & Hamdan, A.R. "Introducing Software Process Assessment and Certification (SPAC) Model." *Proceeding of the 3rd Malaysian Software Engineering Conference (MySec'07)*, Kuala Lumpur, pp 59-63, 2007.
[15] S. Lauesen, & H. Younessi, "Is Software Quality Visible in the Code?", *IEEE Software*, pp. 69-73, July/August, 1998.
[16] G.R. Dromey, "Software Quality - Prevention versus Cure?", *Software Quality Journal* **11:**197-210, 2003.
[17] Pressman, R. *Software Engineering A Practitioner's Approach*. McGraw-Hill International Edition, 2001.
[18] Bertoa, M. F., Troya, J. M. & Vallecillo, A. 2006. Measuring the usability of software components. *The Journal of System and Software* 79: 427-439, 2006.
[19] Yahaya, J.H., Deraman, A. & Hamdan, A.R. "Software Product Certification Model: Classification of quality attributes". *The First Regional Conference of Computational Science and Technology (RCCST 07)*, Kota Kinabalu, 436-440, 2007.
[20] Dekkers, C.A. & McQuaid, P.A. The dangers of using software metrics to (Mis)Manage. *IT Pro* (March/April): 24-30, 2002.
[21] Ortega, M., Perez, M., & Rojas, T. Construction of a systemic quality model for evaluating a software product. *Software Quality Journal* 11: 219-242, 2003.
[22] Yahaya, J.H, Deraman, A and Hamdan, A.R. A model and methodology for software product certification. *Proceedings of the National Conference Software Engineering and Computer System (Nacses07), Cherating, Kuantan, 2007.*
[23] Dromey, G.R. Cornering the chimera. *IEEE Software* (January):33-43, 1996
[24] Eagles. Evaluation of natural language processing systems, Final Report (Sept). http://www.issco.unige.ch/ewg95/[18 May 200], 1995.
[25] Baharom, F. "Satu Model Pensijilan Perisian Berasaskan Penilaian Kualiti Proses Pembangunan", PhD thesis, UKMalaysia, 2008.
[26] Tripp, L.L. Software Certification Debate: Benefits of Certification. *IEEE*, June: 31-33, 2002.
[27] Stafford, J. & Wallnau, K. Is third party certification necessary? *Proceedings of the 4th ICSE Workshop on Component-based Software Engineering:Component Certification and System Prediction*, pp. 13-17, 2001.
[28] Voas, J. "Software's secret sauce: The "-ilities"", *IEEE Computer*, November/December, 14-15, 2004.
[29] Deraman, A., Yahaya, J.H, & Hamdan, A.R. Software Certification: A Continuous Improvement. *The International Journal of Knowledge, Culture and Change Management*, Vol. 8 No. 7:125-133, 2008.

Jamaiah Haji Yahaya is a lecturer in College of Arts and Sciences (Information Technology), Northern University of Malaysia (UUM), Kedah, Malaysia since 2000. Prior that she worked as a system analyst at University of Science Malaysia (USM). She received her Bachelor of Science (Computer Science and Mathematics) from University of Wisconsin-La Crosse, Wisconsin, USA (1986), Master of Science (Information System) from University of Leeds, UK (1998) and PhD in Computer Science from National University of Malaysia (UKM) in 2007. Her research interests include software certification, software quality and software management. Her email addresses are <jamaiah@uum.edu.my> and <jhyahaya@yahoo.com>.

Aziz Deraman a professor in Software Engineering in Department of Computer Science, Faculty of Information Science and Technology, National University of Malaysia (UKM). Currently he is the Deputy Vice Chancellor at University Malaysia Terengganu (UMT), Malaysia. He received his BSc degree from National University of Malaysia - UKM (1982), MSc degree from Glasgow University (1984) and

PhD from University of Manchester Institute of Science and Technology (UMIST) in 1992. His research interests include IT strategic planning, software management and certification, medical computing and community computing. His email addresses are <a.d@umt.edu.my> and <ad@ftsm.ukm.my>.

Fauziah Baharom is a lecturer in College of Arts and Sciences (Information Technology), Northern University of Malaysia (UUM) Kedah, Malaysia. She received her Bachelor of Science from University of Technology Malaysia (1992), Master of Science in Software Systems Technology from University of Sheffileds, UK (1995) and PhD in Computer Science from National University of Malaysia - UKM in 2008. Her research interests are software process improvement, software quality and software evaluation. Her email address is <fauziah@uum.edu.my>.

Abdul Razak Hamdan is a professor in System Management and Science, Faculty of Information Science and Technology, National University of Malaysia (UKM). Currently he is the Dean of this faculty. He received his BSc degree from UKM (1975), MSc degree from University of Newcastle Upon Tyne, UK (1977) and PhD in Artificial Intelligence from Loughborough University of Technology, United Kingdom in 1987. His research interests include ICT & Strategic Policy, Intelligent Decision Support, Medical Data Mining and Jawi Pattern Recognition. His email address is <arh@ftsm.ukm.my>.