The Architecture of an Integrated Support Tool for Software Product Certification Process

Aziz Deraman[1], Jamaiah Haji Yahaya[2]
[1]Vice Chancellor Office
University of Malaysia, Terengganu
Kuala Terengganu, 21030
Terengganu, MALAYSIA
a.d@umt.edu.my

[2]College of Arts and Sciences
Information Technology Building
University of Northern Malaysia (UUM)
Sintok, 06010 Kedah
MALAYSIA
jamaiah@uum.edu.my; jhyahaya@yahoo.com

ABSTRACT: *Previous studies in software certification provide a set of axiom and supporting models for software assessment and certification. Two main researches were undertaken to study the issue of certification and developed models of software certification. These 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 approach. The models have been tested by case study, which were launched collaboratively with industries in Malaysia. We extend this works and develop a more comprehensive and integrated model of certification. A significant advantage of the proposed integrated model is that it allows users to choose and design their own implementation or working model of certification which fit with their organisation's requirements and expectations. This paper focuses on software certification by product quality approach and the architecture of the support tool that needed in the certification process.*

Keywords: Software Product Certification Model, Software Assessment, Software Certification Toolset, SoCfeS, Pragmatic Quality Factors

Received: 12 December 2009, Revised 28 January 2010, Accepted 31 January 2010

1. Introduction

In software certification, studies of 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.

Chinese proverb says that "there is always a first step in a journey of a ten thousand miles". Therefore, our first step in building a software product certification model is to identify the requirements through empirical studies. This task was implemented in our previous works in software certification. Previous works were successfully constructed two models of certification, which by means of end-product approach and development process approach. This work was then continued to develop an integrated and intelligent software certification system that focused on end-product quality approach. This paper presents the development and underlying process of the proposed model and toolset. It starts with review on software certification issues, follows with previous work and research approach. Lastly, this paper presents the architecture of the integrated toolset, the implementation, testing and evaluation, and conclusion.

2. Review on Software Certification

The term certification in general is the process of verifying a property value associated with something, and providing a certificate to be used as proof of validity. A software certification is defined by Jeffry Voas (1999a) as a fact sheet that spells out known software output behaviours (and it could also spell out known internal behaviours). Stanfford and Wallnau (2001) define certification as a process of verifying a property value associated with something, and providing a certificate to be used as proof of validity. ISO defines certification as "a procedure by which a third party gives written assurance that a product, process or service conforms to specified characteristics" (Rae, Robert & Hausen, 1995). Software certification can be viewed in three perspectives: product, process and people. It is also known as the software quality certification triangle (Voas, 1998).

Software certification is still a new concept in Malaysia but becoming increasingly popular in Europe and United States. At the same time many debates on this topic are reported (Kolawa, 2002; Stanford and Wallnau, 2001; Trip, 2002; Voas, 2000a). Software certification is very closely related to software quality. Software certification process is considered as the extended of software testing and quality. In software industry, many complaints about quality of software have been reported over the years. Software quality is claimed to be degenerating steadily. Vendors are accused of delivering software with bugs that need to be fixed (Voas, 2000a). The prevalence of this practice leads to a general perception among clients that the software industry in the country as a whole lacks of standards and mechanism for monitoring or ensuring product quality. The existence of software quality assurance (SQA) team in the organization alone does not guarantee the quality of software being developed by the organization (Jamaiah, Aziz & A Razak, 2006). The involvement of independent body in the assessment of software is believed to be beneficial to the process because it is claimed to be unbiased to the product. The third party assessment is an alternative evaluation toward certifying the product. Another approach of implementing certification process is through the involvement of end users. In this approach user provides information about the particular products to an identified agency. Reliable Technology adopts the approach of software certification method through involvement of end users (Voas, 2000b). One of the disadvantages of this approach is that users might overlook some of the important technical aspect of the software. User might not perceive some of the technical requirements of the software and ignore some of the importance technical criteria of the software during assessment exercise. Other possible approaches in implementing certification are through developer's self-certification by Morris et al. (2001) and verification and validation technique or LoQuso by Heck & Eekelen (2008). LuQuso technique does not include the behavioural and human aspect of quality in the assessment. These methods have their advantages and drawbacks. Another certification approach was developed by SIF Association (2009) that certified products for a particular conformance release. SIF Association or SIFA issues it's specification that will be used as the certification aid. Such aid is called a conformance release. In our research we proposed a pragmatic quality factors as the assessment benchmark and aid which includes behavioural and human aspect of software quality and a collaborative approach of assessment that includes users, developers and independent assessor in the assessment and certification team.

This approach has several advantages over other approaches such as: -

a. Eliminate bias assessment and evaluation of product by including independent assessor,

b. Eliminate unfairness evaluation by including the owner or developer and users of the product

c. Data privacy and confidentiality is protected

d. Accelerate assessment process by conducting the evaluation in the familiar environment to the team members (Jamaiah, Aziz & A Razak, 2008a)

Previous study indicated that most practitioners just followed the practices that have traditionally been used in the organization. Therefore, most of the organizations in the study were facing quality problem. Some of the problems were software that need to be further improvement after delivery, software were not been delivered on time, users unsatisfied with the quality of the software and budgetary issue (Jamaiah et al., 2005). Several factors have been identified that contributed to this problem and some of the problems are lack of quality assurance skill, immature processes, and non-awareness of the evolving technology. But, previous studies in quality and certification believe that good quality development processes will not guarantee the quality of the product. Therefore, assessment and certification of software must be independent from the development process. A western analogy says that "dirty water can run from a clean pipe" and it is true in software development and construction.

3. Our Previous Work: Software Product Certification Model (SCM-*prod*)

The software certification by product quality approach is an acceptable approach of certifying software that operating in certain environment. A western analogy says that dirty water can run from clean pipes is believed to be true as a good software development processes do not guarantee the excellent quality of product. Thus, assessment of end product software must be independent from the development process.

The fundamental software product certification model was developed and known as SCM-*prod*. It was designed based on the following basis:

i. Assessment by independent body is an advantage to the product's owner by conducting unbiased assessment. The independent certification is believed to be the only approach that user should trust and the demands for it are being heard from both publishers and users (Bertoa, Troya & Vallecillo, 2003 ; Voas 1999b). While evaluation by the SQA team in the organization or the owner/users of the product is beneficial because they know well of the software and will reduce the time taken for assessment process. Thus in this research an assessment and certification of software product are conducted through a collaborative perspective approach between the owner/users of the product, developers and independent assessor.

ii. The candidate software product is completed software and is operational in certain environment.

iii. The software quality factors apply in this research are derived from the ISO 9126 model with enhancement characteristics to accommodate other aspects of software quality requirements.

Fig. 1 shows the components of SCM-*prod* model which consists of pragmatic quality factor (PQF), product certification repository, certification representation method, and assessment team. Refer to our previous publication for detail of this model (Jamaiah, Aziz & Abdul Razak, 2008a).

Pragmatic Quality Factor or 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. While the impact attributes deal with how the software react and impact to the environment. These two components of quality produce a balance model between technical requirement and human factor.

The attributes defined in this model are the 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 IEEE software quality metrics framework (IEEE 1993). The measures are the measurable quality aspects in this model and 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. Survey conducted in this research indicated that there are some degrees of importance of each quality attribute and they can be classified into three layers namely low, moderate and high (Jamaiah, Aziz & Abdul Razak, 2007). For each layer, a range of weight factor is

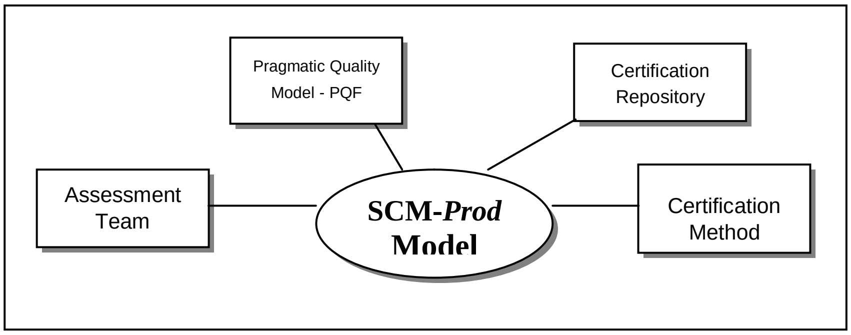


Figure 1. Components of SCM-*prod* Model

assigned and recommended. The beauty of this approach is that software owner 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.

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 (4, 3, 2 or 1) and its representation either of excellent, good, basic and acceptable, or 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.

4. Research Approach

Enhanced model proposed in this research is an integrated model of SCM-*prod* model, which focuses on certification and assessment of software based on product quality approach in a wider scope of requirements. In this model it concentrates on the external quality of the software and may not concern of the development processes. In this relation, we focus on the quality in-use which about measuring the quality of the software in its actual environment. The current research methodology can be described based on the following steps:

4.1 Initial Study: Current SCM Models

This stage involves review and study the current state-of-the-art of Software Certification Models (SCM) in literature. It is also at this stage that a clear understanding of the approach of designing and developing a SCM to be build based on the topic of the subject domain identified, the current limitations and weaknesses of existing methods to help develop an efficient package.

4.2 Toolset Design: Customize and Integrate SCM Model

The design of the appropriate integrate model (i-SCM) for software certification. The design of the appropriate model based on the input obtained from the previous stage. This stage is to develop an integrated model with wider scope of certification requirements.

4.3 Integrated Toolset Development and Implementation

Based upon the input from the previous stage, the development of the i-SCM will be using a systematic design and development approach. The toolset is designed to be flexible, intelligent and capable to learn over time.

4.4 Testing and Refinement I

The developed certification system will then undergo a testing stage to identify any errors and assessing the efficiency and effectiveness of the system. The system will then undergo further refinement based upon the testing results.

4.5 Implementation of the proposed approach for improvement

Once the basic i-SCM have been developed, the research will then implement the proposed approach used to improve its effectiveness based on the refined specifications of the users' requirements.

4.6 Testing and Refinement II

Further testing and refinement on the implemented approach will be undertaken. This will involve industry. This will engage an assessment and certification exercise of a system operated in actual environment.

4.7 Technology Transfer

This will involve conducting a transfer technology workshop to various agencies. Prior to these activities, documentations such as process manual, user manual and system manuals need to be prepared for guidelines in this process.

5. The Architecture of The Toolset - SoCfeS

The fundamental model of software certification or SCM-*prod* model was designed to be used by an authorized body and thus, a more comprehensive and integrated software tools to support the certification processes and environment was required. Ideally, this system supports certification process implemented in any environment. The toolset is named as SoCfeS (**So**ftware **C**erti**fie**r **S**ystem) and it is an integrated software certifier system that consists of an embedded intelligent component

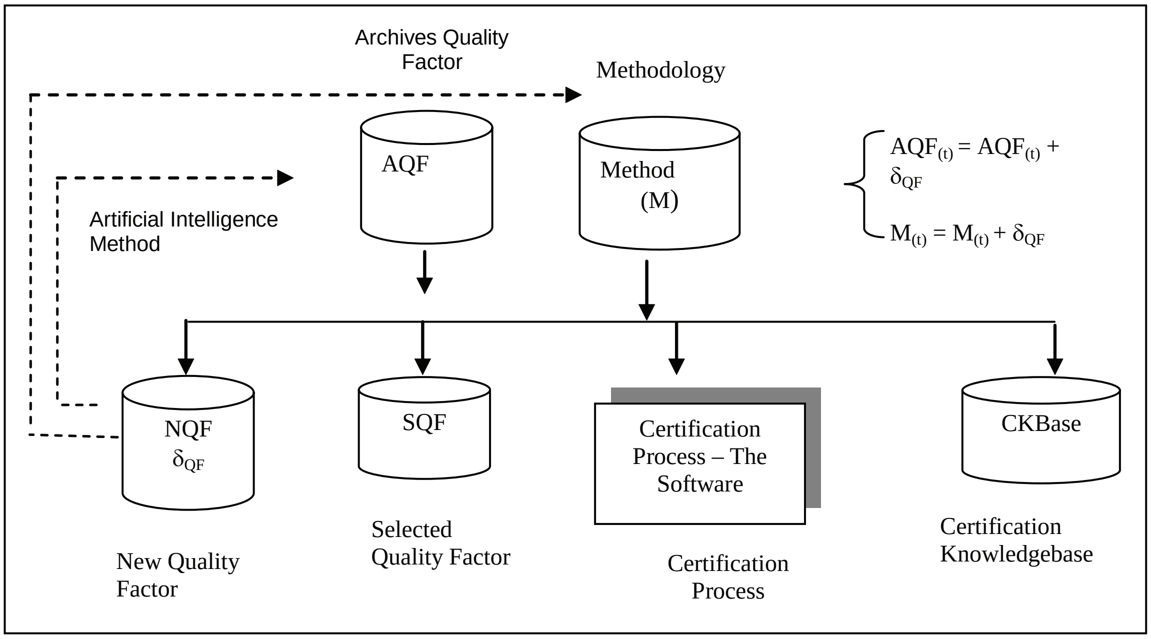

Archives Quality Factor
Methodology
AQF
Method (M)
$AQF_{(t)} = AQF_{(t)} + \delta_{QF}$
$M_{(t)} = M_{(t)} + \delta_{QF}$
Artificial Intelligence Method
NQF δ_{QF}
SQF
Certification Process – The Software
CKBase
New Quality Factor
Selected Quality Factor
Certification Process
Certification Knowledgebase

Figure 2. A Metamodel of SoCfeS

to support the certification environment. The intelligent tool requires a self-learning capability with capturing knowledge from certification processes and experiences. Criteria of software assessment and certification might change from time to time and require additional new criteria to be included in future. The intelligent toolset should capable to notice any changes and therefore recommend to the environment of new modification assessment criteria. The metamodel of the new enhanced certification model is shown in fig. 2.

5.1 The Metamodel of SoCfeS

The metamodel consists of six main components. Each component may represent its own model by itself. The components are:-

- AQF – This represents the achieving quality factors for assessment. It is considered as the master file of quality factors. The notation used is *AQF*. These factors will increase and accumulate from time to time to accommodate pass, current and future requirements.
- Method – Method represents methodology of the certification process. This applies the methodology and algorithm embedded in SCM-*prod* model as discussed in previous section (see also Jamaiah, Aziz & A. Razak, 2008a). The notation used in *M*.
- SQF – It represent the selected quality factor. In this system, users have an opportunity to select their interested quality factors to be applied in the certification and assessment exercise depending on the organizations requirements. Thus, SQF $\in$ AQF.
- CKBase – The knowledgebase of certification. It captures and stores information of certification exercises in various software products intelligently. This component may involve knowledge representation, rules and engineering.
- NQF – NQF represents the new quality factor identified in the environment. NQF is obtained by manipulation of experience and learning capabilities of the system supported by Certification Knowledgebase (CKBase). The notation used is δ_{QF}.
- Certification Process – This represents the system that supports the certification process.

The metamodel above explains briefly on obtaining new quality factors that influence two components which by means of Method and AQF. Therefore, at different time the M and QF are formulated differently as the following :-

$$AQF_{(t)} = AQF_{(t)} + \delta_{QF}$$
$$M_{(t)} = M_{(t)} + \delta_{QF}$$

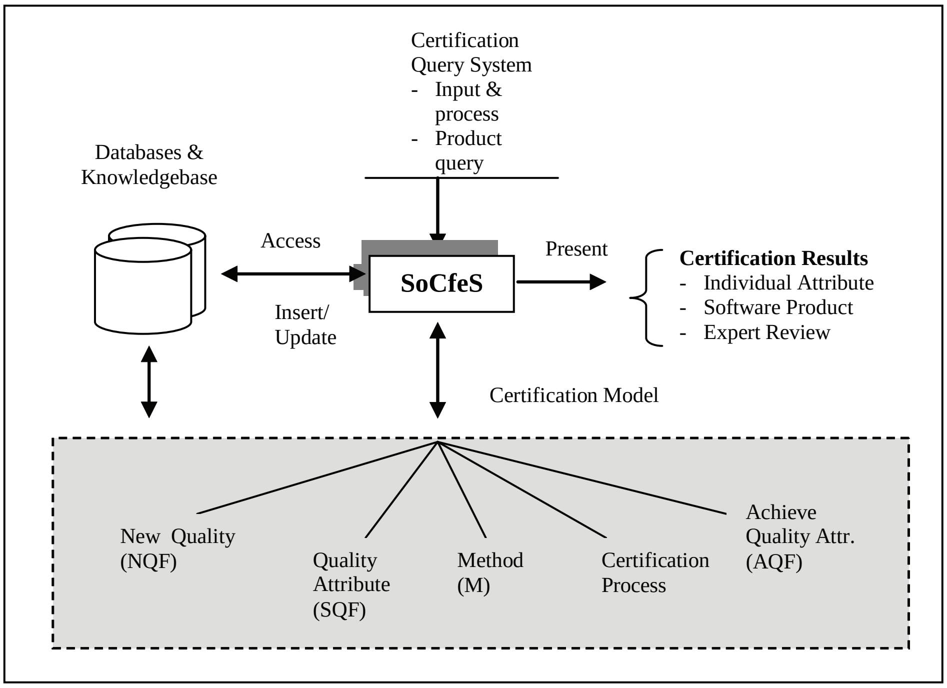


Figure 3. The architecture of SoCfeS

The next step is to represent and transform the metamodel into a system architecture. The data flow of SoCfeS is represented in an architecture that has been designed and shown in Fig. 3. In this first version of SoCfeS it will not cover the intelligent aspect of the model but focuses on the integrated certification process as a whole.

SoCfeS is an integrated system that will use certification data as the input to generate and present the certification results. The certification data is collected through the assessment and certification exercises which conducted collaboratively with other organizations or companies. Once the data has been collected and entered into the system (SoCfeS), it will be stored in a certification database or repository. There will be two levels of data: the raw certification data and the knowledge data. The knowledge data will be captured and processed from previous certification exercises handling by the system. The system will operates based on the certification model and methodology defined in this architecture. The main functions of SoCfeS are capturing and inputting data of certification, certification processing and reporting. The intelligent and knowledge base aspects will be covered in the future version of SoCfeS. It will include the dynamic and intelligent software quality model that will be implemented using an artificial intelligence technique.

5.2 The System Overview

In relation to components in the identified model, SoCfeS requires three main inputs: 1) software product profile, 2) company profile and 3) software product quality data. Product profile needs information on the product such as name, owner, and assessment date and number, and product criteria. A product criterion which includes selection of quality attributes and weight of each attributes needs to be recognized and agreed. A company profile will capture information on the related company that owned the software product. This is necessity to keep track of the certification exercise in the future. It gives information on first certification exercise and renewal version of certification of the product. The third input is the most critical and important input which requires experience in collecting software quality data. The three inputs will be processed to generate and produce the certification reports. At this stage, three certification results will be presented: certification report by individual attributes, certification report by software product and expert review. The quality data mentioned is referred to the Pragmatic Quality Factor (or PQF) in SCM-*prod*. The PQF consists of eight quality attributes which then were broken down into measures and metrics (see also Jamaiah, Aziz & A Razak 2008b).

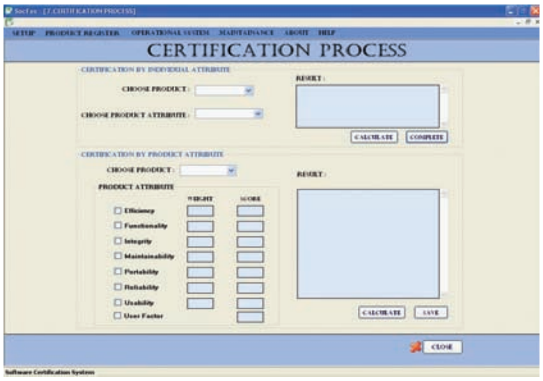
6. The Implementation

SoCfeS system was developed using Visual C# programming tool and MS-Access for database management system. MS-Access is used since its tables are not so large and can easily manipulate from C# through ADO connection. Visual C# is an event-driven, visual programming language in which programs are created using an Integrated Development Environment (IDE). With the IDE, a C# program can be created, ran, tested and debugged conveniently, thereby reducing the time it takes to produce a working program. At least nine classes were built in this project. These classes are: - *Cert_main.cs, ClassEff.cs, ClassFunct.cs*, *ClassMaint.cs, ClassInteg.cs, ClassReli.cs, ClassPort.cs, ClassUsab.cs* and *ClassUser.cs.* The method connects with table *Certification.mdb* that contains all the relevant tables. SQL commands are used to retrieve and manipulate data from tables.

SoCfeS contains several window forms in handling graphical user interface (GUI). It allows users to interact visually with a program. This project contains at least eleven forms and each form handles different tasks. The forms are: *Form_Input-Efficiency, Form_InputFunct, Form_InputInteg, Form_InputMaint, Form_InputPort, Form_InputReli, Form_InputUsab, Form_InputUserF, Form1, Form_Calculate, Form_Weight.*

The implementation has shown that the use of software certification model for assessing and certifying software product is viable. Fig. 4 illustrates a few screen snapshots of SoCfeS.

The whole procedures, processes and support tool in certification were evaluated collaboratively with software industries in Malaysia. This will be explained in the following section.



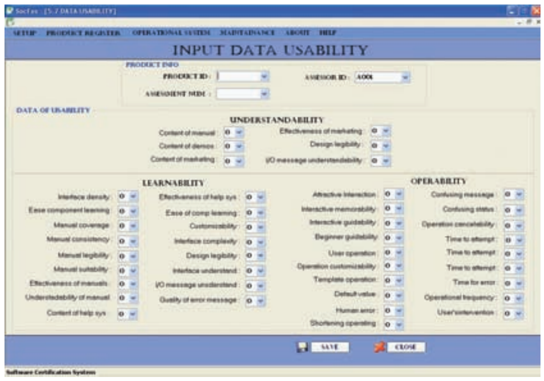


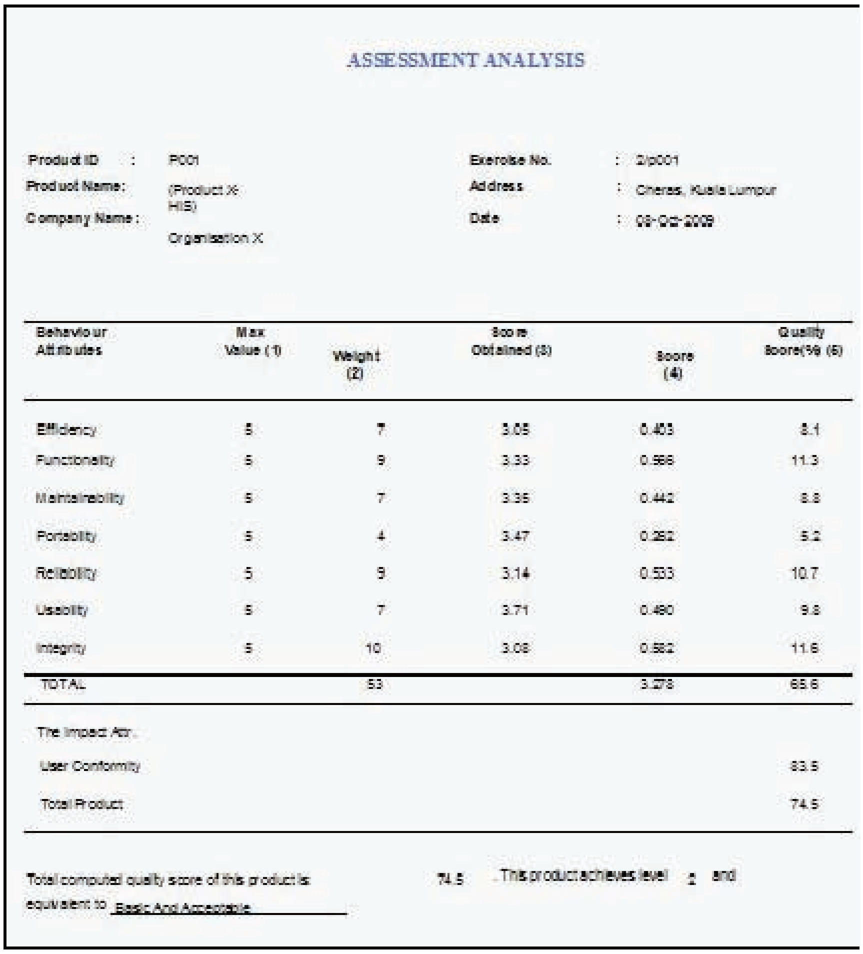
Figure 4. Screen snapshot of SoCfeS

7. Testing and Evaluation

This paper has presented the certification model based on product quality approach which known as SCM-*prod* model. This work was carried out successfully in the first phase of the implementation. This model was considered as a practical model of certification that has been tested and evaluated in real environment. The second phase of the implementation was conducted to improve the underlying processes of certification and specifically in the support tool aspect. Previous version of the support tool did not accommodate enough services for certification process and need wider scope of requirements. Therefore, in this research we proposed an enhanced version of support tool for servicing certification process that included some aspects of intelligent and flexibility in the system. The flexibility of the system was introduced in term of weighting of the attributes and criteria of selection for software quality attributes in the assessment. The two aspects were not detailed in the previous support tool. This is known as the working model of certification which is unique for each organization.

The development of SoCfeS has gone through testing and refinement stages. The first test was the alpha test that was done in-house with internal experts. While the second stage of testing was the beta test which was done collaboratively with Organisation X, involving assessment of hospital information system (HIS) operating in their environment. It is the second exercise of certification on the same software product. The first exercise was done about two years ago. Tab. 1 shows the result from the exercise using SoCfeS which demonstrates that Product X obtains total quality score of 74.5 or equivalent to *Basic and Acceptable*.

The toolset generates a detailed result that consists of score of each attributes and subattributes defined in this model. Calculating them manually is a very messy, tedious and complex task because it involves computation of large numerical data. Tab. 2 shows another example of the result. Results were presented to the stakeholders of this product with some recommendation to improve in regarding to the quality aspect of the system. The detailed results of the case study were documented separately (see Jamaiah, Aziz & Abdul Razak, 2010).



ASSESSMENT ANALYSIS

Product ID :	P001	Exercise No.	: 2/p001
Product Name:	(Product X-HIS)	Address	: Cheras, Kuala Lumpur
Company Name :	Organisation X	Date	: 08-Oct-2009

Behaviour Attributes	Max Value (1)	Weight (2)	Score Obtained (3)	Score (4)	Quality Score(%) (5)
Efficiency	5	7	3.05	0.403	8.1
Functionality	5	9	3.33	0.566	11.3
Maintainability	5	7	3.35	0.442	8.8
Portability	5	4	3.47	0.262	5.2
Reliability	5	9	3.14	0.533	10.7
Usability	5	7	3.71	0.490	9.8
Integrity	5	10	3.08	0.582	11.6
TOTAL		53		3.278	65.6
The Impact Attr.					
User Conformity					83.5
Total Product					74.5

Total computed quality score of this product is: 74.5 . This product achieves level 2 and equivalent to Basic And Acceptable

Table 1. Report on quality score by total product

8. Conclusion

A model that may be used to certify software product has been developed, tested and implemented in real environment. It has been developed in a goal-directed way in order to meet the needs of the different interest groups associated with software quality. We extended and enhanced this model to develop an integrated model that met wider requirements in certification process. This paper explained the architecture and the implementation of software certifier system or support tool. SoCfeS, a software certification support tool, consists of an integrated software certification process. It supports 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. Further more, the toolset was designed to enable users to access their system at their own convenient time. In future version of SoCfeS, it should be able to update the quality attributes and certification component based on the knowledge captured throughout the certification data and exercises. This is important as the quality attributes might change over time based on current requirements and specifications. The new enhanced feature can be supported and implemented by applying artificial intelligence technique.

9. Acknowledgement

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

	Previous (2007)		**Current (2009)**	
Quality Attribute	**Score /5.00**	**%**	**Score /5.00**	**%**
Efficiency	4.08	81.6	3.05	61.0
Time behavior	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	

Table 2. Results in previous and current study

References

1. Bertoa, M.F., Troya, J.M., Vallecillo, A (2003). A Survey on the Quality Information Provided by Software Component Vendors. *In: Proc. of 7th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE 2003)*, pages 25-30.

2. Heck, P., Eekelen, M. v (2008). LaQuSo Software Product Certification Model (LSPCM), http://alexandria.tue.nl/repository/books/633706.pdf.

3. IEEE. (1993). IEEE standard for a software quality Metrics Methodology, http://ieeexplore.ieee.org/xpl/standards.jsp.

4. Jamaiah, H.Y., Aziz, D., Abdul Razak, H (2010). Continuosly Ensuring Quality Through Software Product Certification: A Case Study. *In: Proc. of the International Conference on Information Society (i-Society 2010), London, UK, 28-30 June 2010.*

5. Jamaiah, H. Y., Aziz, D., Abdul Razak, H (2008a). Software Certification Model Based on Product Quality Approach. *Journal of Sustainability Science and Management* 3(2) December 14-29.

6. Jamaiah, H. Y., Aziz D., Abdul Razak, H (2008b). Software Quality from Behavioural and Human Perspectives. *IJCSNS International Journal of Computer Science and Network Security* 8(8) 53-63. August,30.

7. Jamaiah, H.Y., Aziz, D., Abdul Razak, H (2007). Software product certification model: Classification of quality attributes. *In: Proc. Of The First Regional Conference of Computational Science and Technology (RCCST 07), Kota Kinabalu*, pages 436-440.

8. Jamaiah, H. Y., Aziz, D., Abdul Razak, H (2006). Software quality and certification: Perception and practices in Malaysia. *Journal of ICT (JICT),* **5**(Dec) 63-82.

9. Jamaiah H. Y., Fauziah B., Aziz D., Abdul Razak, H (2005). A conceptual framework for software certification. *KUTPM Journal of Technology & Management* **3**(2) 99-111.

10. Kolawa, A (2002). Software certification debate: Certification will do more harm than good. *Computer* 35(6), pages 34-35.

11. Morris, J., Lee, G., Parker, K., Bundell, G.A., Lam, C.P (2001). "Software Component Certification", *IEEE Computer*, September 30-36.

12. Rae, A., Robert, P., Hausen, H (1995). *Software Evaluation for Certification. Principle, Practice and Legal Liability*, McGraw-Hill International.

13. Schools Interoperability Framework (SIF) Association (2009). SIF Certification. Retrieved August 5, 2009, from http://www.softwarecertifications.org.

14. Stanfford, J., Wallnau, K (2001). Is Third Party Certification Necessary? 4th ICSE Workshop on Component-based Software Engineering: Component Certification and System Prediction.

15. Tripp, L.L (2002). Software Certification Debate: Benefits of Certification. *IEEE Software*, June, 31-33.

16. Voas, J (2000a). Limited Software Warranties, *Engineering of Computer Based Systems (ECBS2000) In: Proc. Seventh IEEE International Conference and Workshop*, pages 56-61.

17. Voas,J (2000b). Developing a Usage-Based Software Certification Process. *IEEE Computer*, August 32-37.

18. Voas, J (1999a). Certifying software for high assurance environments. *IEEE Software* July/August 48-54.

19. Voas, J (1999b). Certification: Reducing the hidden cost of poor quality, *IEEE Software*, July/August 22-25.

20. Voas, J (1998). The Software Quality Certification Triangle". CrossTalk, *The Journal of Defense Software Engineering*, November 12-14.

Authors Biographies

Prof Dr. Aziz Deraman, received his Bachelor from UKM in 1982, Master from Glasgow University in 1984 and PhD from University of Manchester Institute of Science and Technology (UMIST) in 1992. He is presently a professor of Software Engineering specialising in software process, management and certification. He has held various academic administrative positions such as head of Computer Science Department (1985-1988), Deputy Dean of IT Faculty (1992-1995), Deputy Director of Computer Centre, UKM (1995-2001), the Dean of the Faculty of Information Science and Technology, UKM, the Deputy Vice Chancellor of University of Malaysia, Terengganu (UMT) (2007-2009) and currently the Vice Chancellor of University of Malaysia, Terengganu since April 2009.

Dr. Jamaiah Yahaya is a senior lecturer at Northern University of Malaysia (UUM), Sintok, Kedah, Malaysia since 2000. Prior that she worked as a system analyst at University of Science Malaysia, Penang. Her bachelor degree was Bachelor of Science in Computer Science and Mathematics from University of Wisconsin-La Crosse, USA (1986), Master of Science in Information System from University of Leeds, UK (1998), and PhD in Computer Science from National University of Malaysia (UKM) (2007). Then she continued her PhD research as a post doctoral fellow in UKM in 2008. Her research interests are software certification, software quality and software management.