The Development of an Integrated and Intelligent Support Tool for Software Certification Process

Jamaiah Haji Yahaya[a], Aziz Deraman[b], Abdul Razak Hamdan[c],
Siti Zahidah Abdullah[c], Fauziah Baharom[a]

[a]College of Arts and Sciences,
Information Technology Building, Universiti Utara Malaysia, Sintok, 06010 Kedah, MALAYSIA
[b]Vice Chancellor Office,
University Malaysia Terengganu, Kuala Terengganu, 21030, Terengganu, MALAYSIA

[c]Faculty of Information Science and Technology,
Universiti Kebangsaan Malaysia
Bangi, 43600, Selangor , MALAYSIA

{jamaiah@uum.edu.my, a.d@umt.edu.my, arh@ftsm.ukm.my, sza42@yahoo.com, fauziah@uum.edu.my}

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 continue to 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 software quality metrics to be updated, modified and added as necessary in the future.

Keywords
Software Certification Model, Intelligent Certification Model, Software Assessment, Software Certification Toolset

1.0 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. Software certification can be viewed in three different perspectives: personnel, process and product and also known as certification triangle (Voas, 1998). 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 first task in building a software product certification model is to identify the requirements through empirical studies. This task was implemented in our previous study in software certification. This study was successfully constructed two models of certification, which by means of product approach and development process approach. This work is then continued to develop an integrated and intelligent software certification model. This paper presents the development and underlying process of the proposed model. It starts with identifying the objectives of this research, and then it discusses issues in literature review, follows with previous work and research methodology. Lastly, this paper presents the architecture of the integrated model and the conclusion.

2.0 OBJECTIVES

The objectives of this study are as follows:-

- To design a comprehensive and integrated model for software certification model based on product quality approach.
- To develop an intelligent support tool for integrated software certification model.

- To provide guidelines and toolkit of integrated software certification process.

3.0 LITERATURE REVIEW

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 (1999) as a fact sheet that spells out known software output behaviours (and it could also spell out known internal behaviours). Stanfford and Wallnau (1997) define certification as a process of verifying a property value associated with something, and providing a certificate to be used as proof of validity. Certification is a means for improving the discipline by promoting the practical implementation of standards, the awareness of a body of knowledge, the recognition of a code of ethics, and the need for professional development (Tripp, 2002). 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). Certification of software can be viewed in three perspectives: product, process and personnel. It is also known as the software quality certification triangle (Voas, 1998).

Many complaints about quality of software have been reported over the years. Software quality are 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 (Voas, 2000b). Other possible approaches in implementing certification are through developer's self-certification (Morris et al, 2001) and verification and validation technique (Heck & Eekelen, 2008). On the other hand, each of these methods has its advantages and drawbacks. In this research we proposed 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)

According to Tripp (2002), certification is a means for improving the discipline by promoting the practical implementation of standards and building awareness of using best development practices. However, previous study indicated that most practitioners just followed the practices that have traditionally been used in the organization (Jamaiah et al., 2005). Therefore, most of the organizations in the study were facing quality problem. Some of the problems are 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.

The Capability Maturity Model (CMM) that was developed by Software Engineering Institute is an example of a mechanism for software quality improvement. It is based on software development process and defines in term of maturity levels. There is another mechanism for improvement and assessment but involves people and skill, examples are People Capability Maturity Model (PCMM) and British Computer Society. Previous researches on product and process quality approaches have been proved to be appropriate solution to certify software product and to guarantee the quality of the software product and satisfy with the contracted agreement between clients and developers

4.0 PREVIOUS WORK

Previous studies were successfully implemented and completed. Two closely related certification models have been developed by this research group: SPAC (Fauziah et al. 2008) and SCfM_*prod* (Jamaiah, Aziz & Abdul Razak, 2008). Both certification models are based on process development approach and product quality approach respectively. The methodology used in the development was divided into 4 phases:-

- Phase 1: Initial requirement study: This phase involved studying and reviewing current state-of-art in the development of certification process model and related subjects. Two separate surveys were conducted to explore the awareness of certification in Malaysia and to understand the requirements of certification. Findings from these reviews were used as the basis for producing the requirement specifications for these models.
- Phase 2: Requirement Analysis and Modeling: This phase of study involved producing specification requirements for two proposed models, product and process approaches.
- Phase 3: Model Construction and Development: There were two methodologies constructed for two different approaches in the certification process.
- Phase 4: Implementation and Evaluation: Two case studies were organized to assess and evaluate the models. The evaluations were carried out with cooperation with Hospital University Kebangsaan Malaysia and Lembaga Tabung Haji Malaysia.

These models were constructed based on findings in empirical studies as well as literature analysis. They were evaluated and tested in two real case studies. The case study indicated that the evaluated aspects of the method were feasible and practical, improved confident and assurance of quality to users and owners of particular product.

Our current developed models participated in solving problem in ensuring and determining quality of software product. SCfM_*prod* model, the model for software product certification has been developed, tested and evaluated but it was developed to work out with current requirements on certification and quality issues.

The quality model or pragmatic quality factor (PQF) applied in certification model (SCfm_*prod*) 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 our model and apply in a more comprehensive and integrated model of certification. This will explain in the following section.

5.0 RESEARCH METHODOLOGY

Current research proposes an integrated model of SCfM_*prod* model, which focuses on certification and assessment of software based on product quality approach. In this model it only concerns with the external quality of the software and may not concern of the development processes. The current research methodology can be described based on the following steps:

5.1 Initial Study : Current SCM Models

This stage involves review and studies the available model of SCM. 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.

5.2 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.

5.3 Intelligent Toolset Development and Implementation.

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

5.4 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.

5.5 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.

5.6 Testing and Refinement II

Further testing and refinement on the implemented approach will be undertaken. This will involve industry.

5.7 Technology Transfer

This will involve conducting seminars, conferences or workshops to various agencies.

6.0 THE ARCHITECTURE OF THE TOOLSET

As mentioned in this paper, the main objectives of this research are to develop an integrated certification model and a toolset to support the certification process.

The first version of certification model or SCfM_*prod* model is designed to be used by any authorized body and thus, a more comprehensive and integrated software tools to support the certification processes and environment is required. This system supports certification process implemented in any environment. We name this toolset SoCfeS (**So**ftware **C**erti**fie**r **S**ystem). SoCfeS is an integrated software certifier system that consists of an embedded intelligent component 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 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 basic data flow of SoCfeS is represented in an architecture that has been designed and is shown in Figure 1.

The components in this architecture 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*.
- Method – Method represents methodology of the certification process. This applies the methodology and model (SCfM_*prod*) discussed in the previous section. 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. SQF $\in$ AQF.
- CKBase – The knowledgebase of certification. It captures and stores information of certification exercises in various software products intelligently.
- 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.

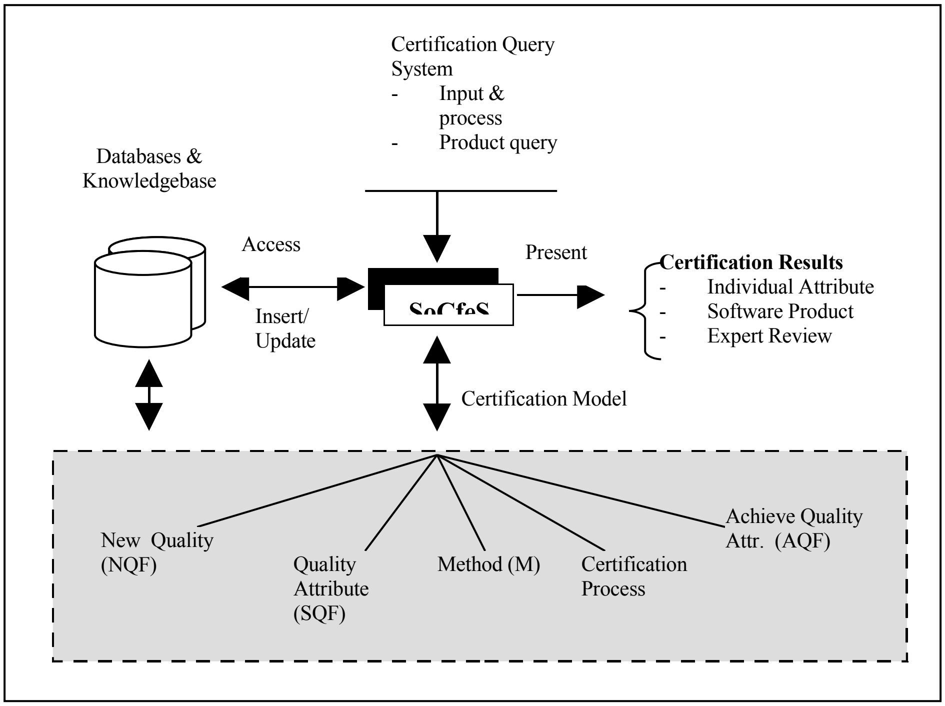

Figure 1: The architecture of SoCfeS

SoCfeS is an intelligent system that will use certification data as the input to produce or present the certification assessment results. The certification data is collected through the assessment and certification exercise which conducted collaboratively with the organisation. Once the data has been collected and inputted into the system (SoCfeS), it will be store 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 from previous certification exercises handling by the system. The system will operates based on the certification model and methodology defined in this architecture. At this stage, three certification results will be presented: certification by individual attributes, certification by software product and expert review.

The intelligent aspect embedded in SoCfeS is through the ability to learn from previous experience. Certification exercises data conducted and implemented using this environment will be kept in the database and will be extracted into a knowledge base. This data will be used to provide new quality attributes relevant and suitable for current and future requirements. Therefore, SoCfeS will recommend to the environment of any new attributes to be included in the assessment and certification. With this ability, the toolset has a capability to improve itself in the environment.

Currently, this research is still in progress. The toolset development, SoCfeS, is almost completed and need to be tested and refined. More works need to be focused in packaging and technology transfer which involves industry.

7.0 CONCLUSION

A model that may be used to certify software product has been presented. This model has been developed in a goal-directed way in order to meet the needs of the different interest groups associated with software quality. The model is a practical model of certification, which was evaluated and tested, in real case studies in Malaysia. We extend this model to produce an integrated model to meet wider requirements in certification process. SoCfeS is a software certifier system developed in this research. The architecture of SoCfeS was explained in this paper. SoCfeS, a certification toolset, consists of intelligent expert system embedded for supporting the continuous improvement in certification environment. It supports this enhance capability via self-learning with knowledge over the certification environment. SoCfeS updates 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 specification.

REFERENCES

Fauziah B, Aziz D, A. Razak H. & Jamaiah H. Y. (2008). Software Certification towards Assuring and Improving the Quality of Software Development Process. *Proceedings of Malaysia Software Engineering Conference (MySEC'08),* Kuala Terengganu, Dec 16-17, 2008.

Heck, P & Eekelen, M. v. (2008). "LaQuSo Software Product Certification Model (LSPCM)" *http://alexandria.tue.nl/repository/books/633706.pdf*. [16 July 2008].

Jamaiah Haji Yahaya, Aziz Deraman & Abdul Razak Hamdan. (2008a). Software Certification Model Based on Product Quality Approach. *Journal of Sustainability Science and Management.* Volume 3 Number 2, December 2008:14-29.

Jamaiah Haji Yahaya, Aziz Deraman, Abdul Razak Hamdan. (2008b). 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.

Jamaiah Haji Yahaya, Aziz Deraman & Abdul Razak Hamdan. (2006). Software quality and certification: Perception and practices in Malaysia. *Journal of ICT (JICT),* **5**(Dec): 63-82.

Jamaiah Haji Yahaya, Fauziah Baharom, Aziz Deraman & Abdul Razak Hamdan. (2005). A conceptual framework for software certification. *KUTPM Journal of Technology & Management*, **3**(2): 99-111.

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

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

Stanfford, J. and Wallnau, K., (1997). "Is Third Party Certification Necessary?. Citing Internet sources URL http://citeseer.nj.nec.com/.

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

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

Voas, J. (2000a). Limited Software Warranties, *Engineering of Computer Based Systems (ECBS2000) Proceedings Seventh IEEE International Conference and Workshop*, pp56-61.

Voas,J. (2000b). Developing a Usage-Based Software Certification Process. *IEEE Computer*, August 2000. pp32-37.

Voas, J. (1999). Certifying Software for High Assurance Environments. Citing Internet sources URL **http://citeseer.nj.nec.com/**