Australian Journal of Basic and Applied Sciences, 7(8): 9-18, 2013
ISSN 1991-8178

Software Certification Modeling: From Technical to User Centric Approach

[1]Jamaiah H. Yahaya, [2]Aziz Deraman, [3]Siti Rohana Ahmad Ibrahim, and [4]Yusmadi Yah Jusoh,

[2]Dept. of Computer Science, Faculty of Science & Technology, University Malaysia Terengganu, Kuala Terengganu, Malaysia
[1,3]School of Computer Science, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi, Malaysia
[4]Dept of Information Systems, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang, Malaysia

Abstract: One of the most essential issues in software products is to maintain it relevancy to the dynamics of the user's needs. Many researches and studies have been carried out in quality aspect of software products to overcome these problems. Several software quality assessment models and metrics have been introduced with strengths and limitations. The current quality models such as McCall, Boehm, ISO9126, Sistemic and PQF model are used as the benchmark for product assessment. In order to enhance the assurance and buoyancy of the software products, certification process and models have been introduced and developed. Previous fundamental and basic software certification models such as SCM-Prod and SPAC were developed to solve the uncertainties in software quality in two views which are the end product and the development process. However, the models are based on specific requirements and have certain limitations. SCM-Prod model focuses on the certification and quality requirements of software products in general. While SPAC model focuses on software certification based on development process approach. Our previous experiences in conducting certification exercises and case studies collaborating with several agencies in Malaysia, the requirements for user based software certification approach is needed and demanded. The emergence of social network applications and other varieties of software in the market has led to the domination of users over the software. As software become more accessible to the public through internet applications, users are becoming more critical in the quality of the services provided by the software. Users should be able to assess and certify their own products within their own environment at anytime and anywhere. The quality attributes for software assessment should be based on user's perspective and view. This new paradigm should the alternative focus in software product quality assessment and certification. This paper presents the evolution of software certification and assessment from technical view to user centric approach. It discusses the concept of user centric approach in software assessment and certification which also focuses on the design and development of the model (ucSoftC), the components and attributes embedded in this model. The development is believed to be beneficial and valuable to overcome the constraints and improve the application of software certification model in future.

Key words: software certification, user centric approach, ucSoftC model, user centric attributes

INTRODUCTION

In general, quality software is refered to the software product that meet its user requirements. Therefore, much concerns of the software community are to find ways on how to establish high quality software processes and how to produce a high quality software product. Existing software processes and products are evaluated and assessed so that the quality level of the software process and product can be determined. Therefore, one of the approach is to certify the software software process and product based on prescribed criteria. Research and development of this effort on software certification can be found in (Deraman & Yahaya, 2012). Continuous improvement in the quality of software is also an essential element in the software industry. As discussed in (Deraman & Yahaya, 2012; Deraman, 2011), software continuous improvement could also be achieved through implementation of software certification within user environment itself.

As the role of software in human life is continuously increased and demanded, software developers are competed to produce software faster and quicker. However, software projects failure are still being reported. According to Ropponen & Lyytinen (2000) and Berntsson-Svensson & Aurum (2006), a software project is considered failed if it is over schedule, over budget, does not meet business objective and does not meet user requirements. From the social and economic aspects, customers or users will lose their confidence while in term of economy; maintenance cost will increase if the project fails.

Corresponding Author: Jamaiah H. Yahaya, School of Computer Science, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi, Malaysia

Our previous works has developed and implemented in real industry a software product certification model named SCM-Prod model which focused on measuring the behavior aspects of the software and limited focused on the user's aspect of measurement (Yahaya, Deraman & Hamdan, 2008a). Past researches and experiences have shown that involvement of users in software assessment and certification process enable to evaluate the quality of software products based on user's expectations and needs. However, SCM-Prod and other certification models do not emphasis on the quality metrics of user centric perspective and approach. Therefore, this approach is an alternative mechanism for assessment and certification to resolve the uncertainties related of software quality in user's perspective. In this research, user centric approach is defined as a method in which the requirements and limitations of end-users of software are given extensive attention at each stage of the processes. Users can be defined as end-users, customers, advanced users, management, public users and etc.

The research is conducted in four main phases as follows:-

- Theoretical Study: The literature review on the existing research related to software assessment and certification is carried out and studied. The aim is to investigate the existing mechanism and problem related to software certification. The detail theoretical aspect of software certification and assessment based on user centric is outlined and important and expected features that contribute to this research are identified and documented.
- Design of theoretical framework on user centric certification: The second phase of this research is designing the theoretical framework on user centric certification. It involves refinement of specific feature of software certification to represent user centric approach. The theoretical framework has been presented in (Ibrahim *et al*, 2012).
- Development of user centric certification model (ucSoftC): The third phase of the research is to develop the model of software certification that focuses on user centric approach. The development will enhance from the recent certification model that is SCM-Prod model.
- Conformation study: The proposed model will be tested and validated in specific software. Feedback from the testing and validation will be used to refine the model. For this purpose, the Web-Based Student Information System of Universiti Kebangsaan Malaysia (UKM SPMWEB) is chosen to be used in the confirmation study.

The rest of this paper is organized as the following: section 2 presents the background and related works, section 3 discuss the design of ucSoftC model and quality attribute of user centric and Section 5 will conclude this paper.

Background And Related Works:

A review on current literature shows that software is a designed not a manufactured. It is something creative that given significant for individual skills and experiences in software development. Novelty of software indicated by its quality without respective of the process used (Sommerville, 2007). Nowadays, software is a tool that people often used and needs to be improved and enhanced its quality frequently (Patton, 2007). In addition, the attributes of software quality is difficult to measure directly such as maintainability, usability and efficiency because it relates to developer and users which they must use the software (Sommerville, 2007).

Software certification is defined as an official document of a good quality. It is also a written assurance by third party organization that a product or services conforms to specified characteristics. Software certification can be viewed in three different perspectives includes through the development process, the end product quality and the people that involve in development (Voas, 1998). Software certification offers benefits and values to several groups include the developer, producer (stakeholder and vendor) as well as the end users. In user perspective, certification is a mechanism to guarantee that the software is good at certain level of quality.

ISO 9126 defines a quality model as a set of characteristics and the relationships between them, which provide the basis for specifying quality requirements, and evaluating quality. Quality factors known as software characteristics such as functionality, maintainability, reliability, usability, reusability and portability (Rawashdeh & Matalkah, 2006). Some researchers view that the quality of a product relates to the process quality. However, the other mentions that relationship between product quality and process quality is more complex in software development. Although, using the software for a long time, software quality still difficult to measure. Consequently, it is hard to predict the process change influence the product quality due to design and creativity roles in software process (Sommerville, 2007). Our previous study has revealed that the defined quality attributes in software quality models are difficult to meet current requirement and specification for software product assessment. This is due to the limitation of the models that consist of static attributes with no capability to learn based on new requirements from the environment (Yahaya *et al*, 2010).

Literature study has come out with some well-known software quality models such as McCall, Boehm, FURPS, ISO9126, Dormey, SQuaRE and PQF (Rawashdeh & Matalkah, 2006; Deraman *et al*, 2010; Panovski, 2008). However, these models mainly focus on technical perspective and limited concern with user's or human's perspective in assessment of software products. For example, ISO/IEC 15504 is the reference model for the process assessment and ISO9126 is the reference model for product assessment. Both reference models focus on technical aspects in development and product quality. Furthermore, K-Model is a software process improvement and certification for small and medium size business and is implemented in Korea (Hwang, 2009). Therefore, in our research, software certification model by user centric approach has been proposed to improve the existing software certification model that meets user's needs and demands.

A. User Centric Issues and Approach:

Today, software is not only vital to the businesses to excel, compete and remain competitive locally and globally but to the social society as a whole. It has become parts of everyone in everyday life. The emergence of social network application such as Facebook, Twitter, Friendster and many more show the relevance and influential of software and computers in todays' people's life. The integration of human's activities and ICT appliances connects people anytime and anywhere through software applications. Thus, this situation create the user centricity paradigm where people and users are the key actors in the scenarios.

User centricity approach has been discussed in several domains and areas especially in software development. It has been highlighted the necessities of user involvement in design or development decisions (Patton, 2007; Quasthoff & Meinel, 2007). Jeff Patton highlighted that productivity of developer has been focused without considering the user's perspective in software development. He suggested that the target of producing software should be building something useful for the people. Normally, in many circumstances, users don't have much concern about the design of the software as long as they can access and use it. Therefore, as a developer they need to understand the wants and needs of user through what they build, who will use it and how to use it. He stated that "user centricity isn't just caring about users or asking them what they want. It's understanding them and collaborating effectively with them to help make informed choices about what software to build". This relates to how to ensure the software meets this target. Thus, the assessment and measurement of software quality must towards the user centricity approach as well.

Previous certification studies by our research group have indicated that certification process demands for self-certifying approach (Yahaya *et al*, 2010). In this new paradigm, users should be able to assess and certify their products within their own environment. In addition, user-centricity approach can be defined as a new paradigm where the requirements and limitations of end-users of software products are given extensive attention at each stage of the processes and user has a control through her involvement in using the product (Spantzel *et al*, 2007). This relates back to the general definition of quality, where quality is defined as "fitness for use" and "conformance to requirements". The term "fitness of use" usually means characteristics such as functionality, usability, maintainability, and reusability and "conformance to requirements" means that software has value to the users (Kim & Park, 2011; Deraman & Yahaya, 2012).

A related work conducted by IBM Switzerland and European Commission IST Project PRIME studied the user centricity concept in federated identity management (FIM) as to provide stronger user control and privacy. In this recent paradigm, two main focuses and being explore are the security and privacy, and the user-control. Under the structure of user control there are properties that relevant to it. The properties such as confidentiality, integrity, revocability, unlinkability, policy, user-chosen IP, verifiability, generated tokens, illegal sharing prevention, non-transferability, and non-replay are identified and used. There are such attributes that considered as high level properties such as accountability, notification, anonymity, data minimization, attribute security, and privacy (Spantzel *et al*, 2007).

According to Quasthoff & Meinel (2007), " User centricity in identity management systems does not only refer to design processes leading to better usability, customer satisfaction or something similar". Ahn *et al* (2009) studied the user centric approach in identity management which focused on user perspective related to managing private and critical attributes. User is able to control their rights and responsibility over the identity information. Thus, the user's private information is better protection by user itself. In this approach, user is an important element to put them into the middle of transaction between identity providers and relying parties. Furthermore, user centricity in healthcare identity management is introduced to for improvement of healthcare and services and the reduction of costs. Patient is the real users of an electronic healthcare infrastructure. While, health profesional such medical practitioners and pharmacists are users of the system. Thus, it will be user centric with health infrastructure, patients, health professional and administrative personnel (Quasthoff & Meinel, 2007).

B. Software Certification by Process:

Software Process Assessment and Certification Model (SPAC) was developed to ensure the effectively of software development process to meet the expected quality criteria, delivered on time and within budget. Basically, SPAC consists of seven components which are the candidate software, the process quality factor, the certification and quality index, assessment team and repository (Yahaya *et al*, 2009; Baharom *et al*, 2011).

The first component of SPAC is the Software Process Quality Factor (SPQF) that measures the quality of software process. The five factors defined in SPQF are includes process, people, environment, development technology and project constraint. The second component is the candidate software that is a completed software product and is readied to be delivered to users or customers. The assessment team is the third component that consists of developers, independent assessor and project manager. The fourth component is the assessment and certification process. There have three main phases of implementation includes preparation, execution and post assessment phase. The fifth and sixth components are the quality and certification level. The seventh component is the repository which is all information and results data from assessment and certification exercises will be store for future analysis and improvement (Yahaya *et al*, 2009; Baharom *et al*, 2011).

Besides SPAC, the Capability Maturity Model (CMM) was the earlier software certification by process. CMM was developed to evaluate and maintain the maturity capability of the organization. Thus, a guideline has been prepared to gain control of the process. Therefore, the selection of process improvement strategies was important for organization software to determine the process maturity. Besides, CMM is used to identify the improving of processes and issues pertaining to software quality (Paulk *et al*, 1993).

There are 5 levels defined in CMM which are initial, repeatable, defined, managed and optimizing. The first level is initial, which the process of software is ad hoc. It caused of changed or modified during the work progress. Thus, the software process such as scheduling, budget, functionality and product quality are unpredictable. The second level is repeatable, which the organization can repeat their earlier successful of project. In this level the organization already has a stable plan and track in previous project. The third level is defined that involves documented, standardized and integrated organization standard of software process. The fourth level is managed that the organization sets the quantitative quality for software process and product quality. The fifth level is optimizing that focuses on continuous process improvement to identify the weaknesses and strengthens of the software process in organization (Paulk *et al*, 1993).

C. Software Certification by Product:

Software Product Certification Model (SCM-Prod) is a certification model which focuses on assessing software by end-product quality approach. SCM-Prod model consists of six main components includes Pragmatic Quality Factor (PQF), product criteria, certification team, certification specification, certification representation method and product certification repository (refer Fig. 1) (Yahaya *et al*, 2008a; Deraman *et al*, 2010; Yahaya *et al*, 2012).

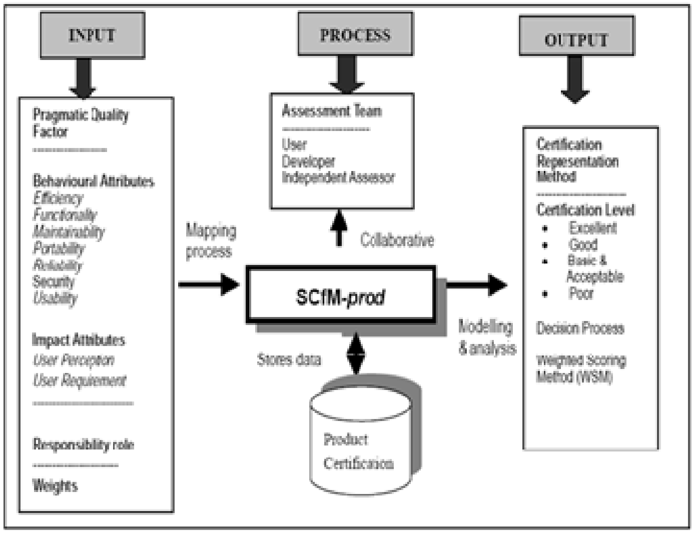

Fig. 1: Software Certification Product (SCM-Prod) model (Yahaya *et al*, 2008; Deraman *et al*, 2010)

The first component is Pragmatic Quality Factor (PQF) that a quality certification guidelines and standard to measure software quality product. The qualities attributes are divided into two set of attributes: the behavioural attribute and the impact attributes. The quality factor in this model is derived from ISO9126 model and enhanced with several additional attributes relevant for current requirements. It is based on our empirical study conducted in Malaysia. The survey was conducted with agencies in Malaysia to identify and verify the suitability of factors for software product assessment (Yahaya *et al*, 2006) particularly for certification model. The second component is product criteria that offer service of weight factors and the criteria selection, which user can select their own quality attributes to certify the software product. The third component is certification team that involves in the certification process which consists of the independent assessor, developer and users. The fourth component is certification specification that describes the processes, algorithm, formula and reporting format in the certification model. The fifth component is the certification representation method that determines quality status and certification-mapping process to obtain certification level of software product such as excellent, good, basic & acceptable and poor. The last component is the product certification repository that will store all data and reports. SCM-Prod model is supported by a tool (SoCfeS) which automates and simplifies the underlying processes and procedures (Deraman & Yahaya, 2011). SCM-Prod model has shown the capability to be implemented in real working environment (Yahaya *et al*, 2008b; 2010).

LaQuSo is another example of software certification model by product that used verification and validation technique to certify the software. The LaQuSo certificate can be requested by organizations for certainty or confidence in software artifact. Certification of LaQuSo will check the artifact based on the fulfilled and defined requirements. In this model, the certification covers only on product quality and not involves the management and the development process. LaQuSo is the independent evaluator in software certification projects (Heck & Eekelen, 2008).

Previous issues in software certification led to the development of software certification model by independent and third party assessment such as by Bertoa *et al*, Standfford & Wallnau and Jeffrey Voas (1998). A few countries have started to develop the software product certification program which involves third party certification body. Korea is one of the leading countries that develop a certification program called Good Software for Korea software industry (IT Times, 2011).

The above discussion has highlighted the benefits of using software certification approach for ensuring quality of software products. It reviews a range of previous method and approach for measuring software performance and quality. The era of measuring quality through internal metrics may not be relevant, complicated and inefficient. Therefore, measurement process now focuses more on external metrics. The evolution of software and software quality and assessment models can be seen in Fig 2.

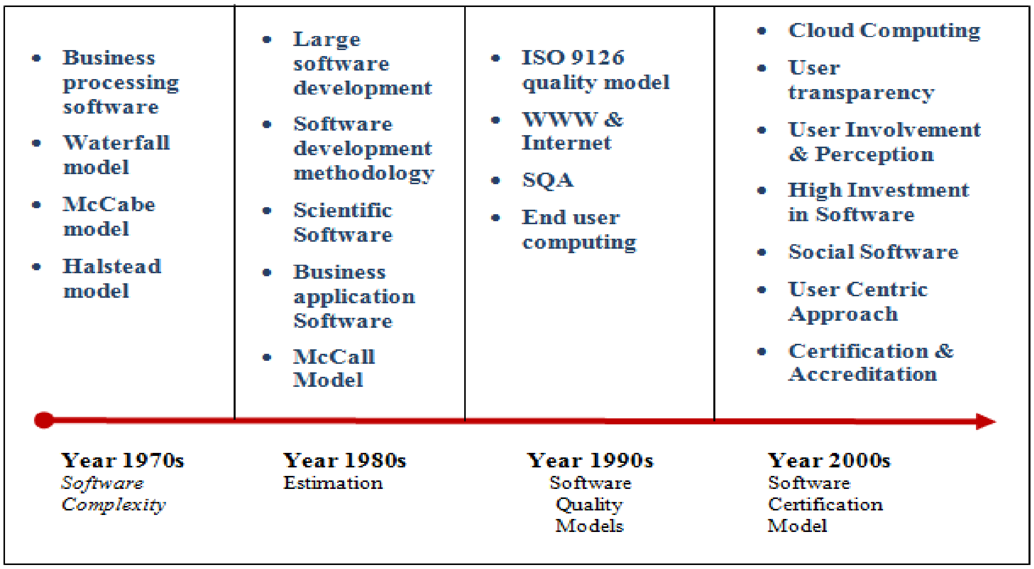

Fig. 2: The evolution of software and software quality and assessment model (Deraman & Yahaya, 2012)

The evolution of software product and software quality and assessment methods can be summarised from year 1970s to years 2000s. It shows the development of software assessment method from measuring through complexity, estimation and later moves to the development of software quality model such as McCall and Boehm model. In year 1990s, ISO 9126 was developed which demanded from the industry to measure software based on end-product quality approach. While in year 2000s onward, we can see the emergence of cloud computing, social software and user involvement in software development where the evolution of software certification is needed for assurance of quality. Furthermore, with the domination of users in software applications, there is a demand for user based quality assessment and certification in software industry.

User Centric Software Certification Model (Ucsoftc): The Design:

This section presents the design of user centric software certification or ucSoftC model. It is designed and developed based on current model of software certification which is the SCM-Prod model. The new enhanced model of certification consists of four main components; quality factor (ucQF), certification assessor, certification process and certification level (refer to Fig. 3).

The first component is the quality factor which is divided into two set of attributes, the behavior or technical attribute and user centric attribute. The second component is certification assessor that involve in the assessment. In this model only user will do the assessment of the software product. The third component is certification process which includes the procedures, formulas and algorithms to implement the model. In this model, the computation of quality score then will mapped into the certification level which determine as excellent, good, intermediate/fair or poor. The basic algorithm and formulas are derived from our previous works (Yahaya *et al*, 2008a; Deraman *et al*, 2010).

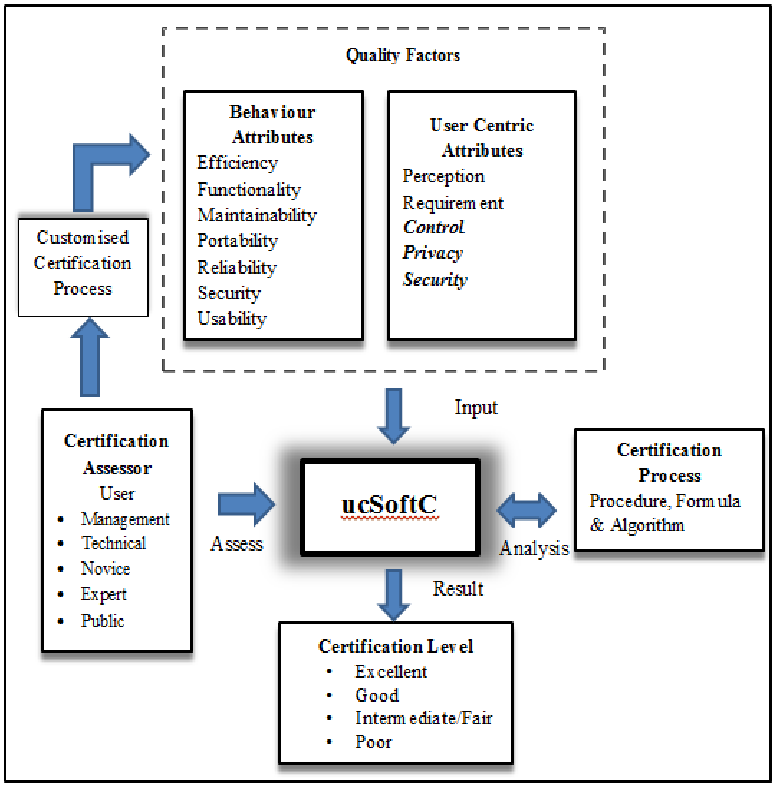

Fig. 3: User Centric Software Certification (ucSoftC) model

The user centric approach in this research refers to the concept of assessment carried out by users only and therefore the quality attributes of assessment mainly focuses on the user centric approach and perspective. In the previous SCM-Prod model, the assessment is carried out by three different groups collaboratively which are the developer, independent assessor and user. In ucSoftC model, new quality attributes are introduced such as control, privacy and security. These attributes represent the user centric characteristics and metrics. Thus, the development of user centric software certification model is an alternative mechanism to enhance and improve the SCM-Prod model for software product certification process.

Fig. 3 shows that depending on the category of users, the certification process which includes the quality factors can be customized to make the certification more practical and reliable to the organization.

D. Certification Assessor:

In this model, the main participants in the assessment and certification are the users of the system software. The identified and potential users of system software are the management, technical, public, novice, and expert. The classified users above have a personalized quality factors relevant and suitable for the group. There is a possibility of customized process of certification and assessment based on user's groups as shown Fig. 3.

E. The Certification process:

The second component is the process. In this research, the formulas and algorithms for user-centric self-certification and assessments are introduced. These formulas and algorithms are used to measure the quality status of each attributes defined ucQF as well as the certification level of the product. The quality score obtained in this model is then mapped into a certification level of excellent, good, basic and acceptable and poor as defined in SCM-prod model. The basic algorithms and formulas of certification are validated and applied in various exercises conducted in Malaysia (Yahaya, Deraman & Hamdan, 2008a, 2010) and needs to be customized to user centric approach. This will be carried out in this research.

The attributes selected and processes defined in this model need to be documented and stated either through hardcopy or softcopy kits and instruments. The assessment instruments need to be checked for verification and validation through expert review and case study. The support system will be developed to assist in the processes and assessment. The proposed technique is through online assessment which is developed using android programming and handheld technology. This is to provide a demanded mechanism where users are able to assess and certify the products in their own environment and within their own sufficient time. With this capability, the method of assessment and certification will be different from other approaches.

F. The Quality Attributes based on User Centric Approach (ucQF):

Several attributes such as control, privacy and security are appropriate to be adopted in user perspectives with relevance metrics. In this structure, the attributes are broken down into several metrics and measures where each measure is scaled into a Likert value of 1 to 5. Table 1 presents the attributes and metrics of ucQF defined in this model.

Table 1: The Attributes And Metrics Of Ucqf

Attributes	Metrics
User Perception	Popularity Performance
	Law & Regulation
	Recommendation Trustworthiness Requirement & Expectation Environmental adaptability
User Requirement	User acceptance Satisfaction
User Control	Accessibility Notification Portability Availability
User Privacy	Policy Data protection Accuracy
User Security	Data confidentiality Integrity Data encryption

The following tables, Table 2 to Table 6, present the attributes, metrics and measurement of each of user centric quality attributes. Table 2 shows the metric and measures of user perception attribute. Table 3 shows the metrics and measures for user requirement attribute; Table 4 defines the metrics and measures for user control while Table 5 and Table 6 define the user privacy and user security respectively.

The quality attributes of software have been identified as shown in previous tables (Table 2 to Table 6). The attributes are derived from the literature and previous studies conducted by this research group. This follows with confirmation study which is carried out with selected software system operating in our environment. The case study is conducted to evaluate software named Web-based Student Information System for Universiti Kebangsaan Malaysia. The case study is still on-going and the findings will be presented and published when the report are readied.

Table 2: The Metric And Measures Of Ucqf: User Perception

Metrics	Measurements
Popularity	- What scale do you measure the popularity of this product?
Performance	- In general, how do you assess the performance of this product? - How do you measure performance based on dependability, efficiency, usability and fidelity?

	- How do you measure the usability of this product? - Do you feel comfortable when interaction with product?
Law & Regulation	- Does the software comply with laws and regulations of your organization?
Recommendations	- Do you recommend this product to your friends?
Trustworthiness	- How do you measure trustworthiness in term of vulnerability (integrity, confidentiality, survivability) and accountability (auditable)?
Requirement & Expectation	- Does the software comply with your requirement and expectation?
Environmental adaptability	- How do you measure the adaptive of this software product in term of interoperability (compatibility and openness), portability, scalability, reusability?

Table 3: The Metric And Measurements Of Ucqf: User Requirement

Metrics	Measurements
User Acceptance	- Do you think this product a useful and ease to use? - Do you feel this product is user-friendly? - How often do you use this product? - Do you think that this product is important for you?
Satisfaction	- What is your rating scale of satisfaction on this product? - How often does discretionary decision being used when using this product? - How do you measure the frequency of complaints in handling the product? - All the functionality was relevant and useful? - How do you measure satisfaction in term of responsiveness, effectiveness, correctness and variability? - In general, do you satisfy with the interface design of this product (design, text and graphic)? -

Table 4: The Metric And Measurements Of Ucqf: User Control

Metrics	Measurements
Accessibility	- Do you agree with the language as provided in this system? - Is the language give benefit to you to access this system? - Do you know if the system has links to other sites? - Is the link give benefit to you while using this system? - How do you measure the accessibility of this system (easy to manage, flexible and more options)? - Do you know your access rights on this system? (read, view and print) or (add, delete and update)
Notification	- Do you know the news or information that displayed in this system? - Have you received any notification from the system through other sources (e-mail, SMS)?
Portability	- Do you use this system through various devices (PC, laptop or smartphone)? - How do you measure this system displays in different web browser?
Availability	- Is that system ready for use at all times (out of a power outage, hardware failure or upgrade the system)? - Do you agree if the system is delayed or downtime for some reason? - Is that data or information can be accessed through the system at all times?

Table 5: The Metric And Measures Of Ucqf: User Privacy

Metrics	Measurements
Policy	- Do you know the privacy policy that used in this system? - Do you think the system requires a privacy policy? - In your opinion, does the encryption necessary to protect the data in this system?
Data protection	- Do you know the encryption technology used to protect the data? - Do you think your personal information (name, address, e-mail or phone number) is protected from unauthorized access? - Does the system provide automatic function log off if no activity is detected after a limit of set time?
Accuracy	- How often do you update the personal information (address, phone number)? - Do you think the data in the system accurate, complete and update?

Table 6: The Metric And Measures Of Ucqf: User Security

Metrics	Measurements
Data confidentiality	- Have your personal information being disclosed, or copied by unauthorised people? - Is that appropriate action should be taken to individual who disclosure information in this system? - Do you agree if personal information (name, address and telephone number) is used to the other party for a particular purpose?
Integrity	- Have your personal information being modified by the other party? - Do you think to access the system using username and password as authentication is adequate the control of system?
Data encryption	- Do you know the security technology used for data encryption in this system? - Do you think the username and password should be encrypted?

G. The Certification Level:

The last component of ucSoftC model is the certification level. The quality score calculated is mapped into a certification level of excellent, good, intermediate or poor. The level represents the quality status of the particular product. This can be used as the indicator of the quality status of the product.

Conclusion And Future Work:

This paper has presented the issues of software certification practices related to software quality assessment. Before advancement of internet technology, most of the researchers focused on software complexity measures which would determine the software quality. With the current development of internet applications and technologies, software applications are more transparent and much closer to the end-users. Software development cycle is also shorter which demand more active user involvement in the process. With the current scenarios in software development and implementation, users are more critical in various functional and non-functional characteristics of the software. For numbers of years, our research group has focused on software quality from different perspective which is through external assessment view which is via certification. We have successfully developed and implemented two certification models as discussed in this paper.

We further our research to focus on user based software certification model. The components and attributes of software certification model based on user-centric approach have been presented. The user centric software certification (ucSoftC) model is a new model to enhance previous software product certification and assessment models explained in this paper. Generally, user centric approach focuses in user perspective to assess and certify software product operating in their environments. Therefore, the developments of user centric software certification process and ucSoftC model are able to fulfill the requirement of organization according to demands and constraints in software product quality and assessment. The proposed ucSoftC model explained in this paper will improve the certification process which enables software users to assess and certify their own products in their own environment with customized and selected attributes based on the organization's requirements and expectations.

REFERENCES

Ahn, G.J. *et al*., 2009. Privacy-enhanced User-Centric Identity Management, *IEEE Communications Society (IEEE ICC) proceedings.*

Baharom, F., J.H. Yahaya, A. Deraman, & A.R. Hamdan, 2011. SPQF: Software process quality factor for software process assessment and certification. *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics,* IEEI 2011, art no 6021526.

Berntsson-Svensson, R. & A. Aurum, 2006. “Successful Projects and Products: An Empirical Investigation” ACM Proceeding of ISESE'06, Rio de Janeiro, pp: 144-153.

Deraman, A., A.R. Hamdan, J.H. Yahaya, F. Baharom, & Z. Hood, 2010. Software Certification Process by Product Quality Approach: The Process Manual, Kuala Terengganu: *Penerbit Universiti Malaysia Terengganu*.

Deraman, A. & J.H. Yahaya, 2011. The Architecture of an Integrated Support Tool for Software Product Certification Process, Journal of Information & Systems Management, 1(1): 46-56.

Deraman, A., 2011. Memburu Kualiti Perisian, *Penerbit UKM: Bangi*. ISBN 978-967-942-967-1, pp: 1-76.

Deraman, A. & J.H. Yahaya, 2012. Research and Development in Software Certification: Past, Present and Future (Keynote), Proceedings of the International Conference on Information Technology and Applied Mathematics 2012, Jakarta, Indonesia, 6 Sept 2012, pp: 1-7.

Ibrahim, S.R.A., J.H. Yahaya, & A. Deraman, 2012, Software Certification Model based on User Centric Approach: The Theoretical Framework, *Proceedings of Malaysian Software Engineering Conference (MySec’12),* UKM Bangi, 5-6 Dec.

IT Times, 2011. “30 Korean Software Worthy of Global Recognition in 2012”. http://www.koreaittimes.com/story/15607/30-korean-software-worthy-global-recognition-2012 (Accessed: 22 August 2011).

Hwang, S.M., 2009. Process Quality Levels of ISO/IEC 15504, CMMI and K-model, *International Journal of Software Engineering and Its Applications,* 3(1): 33-42.

Heck, P. & M.v. Eekelen, 2008. LaQuSo Software Product Certification Model (LSPCM).

Kim, J. & Y. Park, 2011. Two Directions of User-centric Approach to Identifying New Service Opportunities: Vacuum to Solution and Practice to Niche, *3rd International Conference on Information and Financial Engineering IPEDR,* 12: 487-491.

Quasthoff, M. & C. Meinel, 2007. User Centricity in Healthcare Infrastructures, Internet Technologies and Systems Hasso Plattner Institute, Germany, pp: 141-151.

Patton, J., 2007. “Understanding User Centricity”, *IEEE Software,* 24(6): 9-11

Paulk, M.C. *et al*. 1993. Capability Maturity Model for Software, Version 1.1, Technical Report CMU/SEI-93-TR-024 ESC-TR-93-177.

Panovski, G., 2008. Master's thesis Product Software Quality, Department of Mathematics and Computing Science, Eindhoven University of Technology.

Rawashdeh, A. & B. Matalkah, 2006. A New Software Quality Model for Evaluating COTS Components, *Journal of Computer Science* 2(4): 373-381, ISSN 1549-3636.

Ropponen, J. & K. Lyytinen, 2000. "Components of Software Development Risk: How to Address Them? A Project Manager Survey," IEEE Trans. Software Engineering, 26(2): 98-112.

Spantzel, A.B., J. Camenish, T. Gross, & D. Sommer, 2007. User Centricity: A Taxonomy and Open Issues, Department of Computer Science, Purdue University, IBM Zurich Research Lab, Switzerland.

Sommerville, I., 2007. Software Engineering 8, England: Pearson Education Limited, pp: 516-656.

Voas, J., 1998. The Software Quality Certification Triangle. Crosstalk, *The Journal of Defense Software Engineering*, pp:12-14.

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

Yahaya, J.H., A. Deraman, & A.R. Hamdan, 2008a. SCfM_Prod: A software product certification model", *3rd International Conference on information and Communication Technologies:* From Theory to Applications, ICTTA, 2008, art 4530350.

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

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

Yahaya, J.H., A. Deraman, & A.R. Hamdan, 2010. Continuously ensuring quality through software product certification: A case study. 2010 International Conference on Information Society, i-society 2010, London. Art No. 6018821, pp: 183-188.

Yahaya, J.H., A. Deraman, F. Baharom, A.R. & Hamdan 2012. Aligning Software Certification Skill Based on Industrial Issues in Software Engineering Curriculum, *Proceedings of the World Congress on Engineering 2012, London, U.K,* Volume II WCE 2012.