A Case Study of Software Process Assessment and Certification (SPAC) Model Implementation

Fauziah Baharom[1], Aziz Deraman[2], Abdul Razak Hamdan[3] and Shafinah Farvin Packeer Mohamed[4]

[1,4]*Universiti Utara Malaysia, Malaysia, fauziah@uum.edu.my and shafinah@uum.edu.my*
[2]*Universiti Malaysia Terengganu, Malaysia a.d@umt.edu.my*
[3]*Universiti Kebangsaan Malaysia, Malaysia arh@ftsm.ukm.my*

ABSTRACT

This paper reports the implementation of Software Process Assessment and Certification (SPAC) model. In year 2006, the model has been validated in three software organizations for its practicality in the real world. Recently in year 2011, the SPAC model has been implemented again in one of the organizations participated in year 2006. This paper discusses the outcome from the current study and compares it with the previous study. It reveals that after five years, the level of certification decreases from LEVEL IV to LEVEL II. This is because the best practices of software development are being neglected. Thus, we conclude that continuous software certification is certainly needed in order to know the current status of software development process and help the organization to plan and monitor their continuous improvement of software quality.

Keywords: Software Process Certification, SPAC Model, Certification Yardstick.

I INTRODUCTION

Certification is proven as a mechanism that able to give confidence to customers about the Quality Level of a certain product. Along with the increasing software usage, customers are now more concerned on the quality aspects (Pikkarainen, 2009; Heck, Klabbers & Eekelen, 2010). Due to that, many studies have been conducted in the field of software quality and certification by introducing several approaches and models related to software certification. Software certification is defined as a procedure or process where third party identifies key features of the product, process or service and gives assurance that those features and specifications comply with their benchmark (Vermesan, 1998; Rae, Robert & Hausen 1995; Cleland et al., 2003). From these definitions, it is clear that in order to implement the certification process, we need to clearly identify the object to be assessed, the technique to be used and the people who will involve in the process. According to Voas (1998), certification in the software industry can be implemented in three approaches which are people, product and process.

Even though many researchers believe product-based approach can give confidence to consumers about the quality of software (Jamaiah, Aziz & Abdul Razak, 2007; Voas, 1999), at the same time, they admit that quality assessment for product-based approach is hard to be practiced especially for the new software which is just ready to be released. Thus, based on the Deming's premise that "the quality of product is largely governed by the quality of process used to develop it", this study believes that process-based software certification can be alternative solution to determine the quality of software.

Several studies were intended to produce models and standards for software process improvement (SPI) including ISO/IEC 15504 (Pyhajarvi & Rautiainen, 2004; O'Regan, 2002; Wang et al., 1997) and Capability Maturity Model (CMMI Product Team, 2010). On the other hand, the ISO 9000 (Sedani & Lakhe, 2009; Cianfrani, Tsiakals & West, 2009) provides a mechanism to certify only on the quality system of an organization. Besides, the Software Process Assessment and Certification (SPAC) Model which introduced by Fauziah (2008) mainly focuses on certifying software development process in order to ensure that the process was carried out effectively and efficiently. This paper will discuss about the implementation of SPAC Model in the software industry through case studies and compare the results.

II OVERVIEW OF SPAC MODEL

The SPAC Model is a process based software certification model. The model was formulated by referring to existing models or standards which are: Capability Maturity Model (CMM), ISO/IEC 15504 (also known as SPICE), ISO 9000:2000 and ISO 9000-3, and Bootstrap. The SPAC model consists of several main components which are the certification criteria known as Software Process

Quality Factor (SPQF), the SPAC method, certification object, certification technique, certification team, and certification yardstick. It is mainly focused on assessing and certifying the quality of software development process. The SPQF is a goal oriented reference model which defines "what" need to be assessed. Basically, the certification is focused on five factors that influence software quality, which are: process, people, development technology, working environment and project constraints. Certification yardstick contains two main entities to represent certification results which are referred as Quality Level and Certification Level. Interested readers are directed to the previous paper discussing about SPAC Model for further understanding about the model (Fauziah, Jamaiah, Aziz, Abdul Razak, 2011).

III CASE STUDY PROFILE

The SPAC Model has been applied through case study approach in Organization X, which is a computer centre located in Malaysia. The first case study took place in year 2006 while the second case study was conducted in 2011. For the first case study, the development process for developing an e-Academic system was assessed, while the second case study assessed the development process for an accounting and finance system (to be referred as AF System). This paper focuses on the outcome from the second case study and compares it with the outcome from the previous case study. AF System was developed using Rapid Application Development (RAD) approach and utilized Unified Modeling Language (UML) for representing the system requirement and design. It was primarily developed by using INGRES in year 1991 and later in year 2001, this system was upgraded to SYBASE. AF System has 16 modules and being updated from time to time according to the request of the university's bursary (the user of the system).

IV CASE STUDY IMPLEMENTATION

The case study which was conducted from 17 September 2011 until 27 September 2011 involved three phases, as suggested by the SPAC Model. They are discussed below:

A. Phase 1: Pre-Assessment

In this phase, two briefing and discussion sessions were held. The first meeting was between researcher and the Head of Information Technology Officers while the second meeting was carried out with the person in charge with AF System. Both meetings were intended to give some briefing on how the assessment will be conducted, briefing about the SPAC Model and its implementation and objective of the assessment.

B. Phase 2: Implementation

The implementation phase involve with three techniques for assessment, which are document review, interview and observation, as discussed below:

i. Document review: among the documents which were assessed are:
   - System specification requirement
   - System design specification
   - Test forms
   - User manual
   - Document of standard for software development

ii. Interview: three interview sessions were conducted along the assessment period:
   - Interview session 1: involved the project leader whereby it is aimed to get further details and clarification about the software development process applied in the Organization X. This is because most of the processes were not well documented.
   - Interview session 2: involved the software developers which intended to get more details on the practices of software development. Furthermore, their satisfaction level on the organization and trainings provided also were discussed.
   - Interview session 3: involved the user of the system, which is aimed to know customer's satisfaction and the commitment given by the development team.

iii. Observation: researcher has observed the working environment where AF System was developed. It can be concluded that the working environment is stable, conducive and secured.

C. Phase3: Post-Assessment

This phase involved the process of analyzing the data gathered during the assessment, whereby the Score Average (SA) for each of quality attributes was calculated. The value is used for determining the Quality Level. At the end, the Cumulative Score Average (CSA) was calculated to determine the Certification Level. At the end of this phase, a presentation regarding the assessment result was held among the researcher and the involved parties in the assessment. This presentation is aimed to present the outcome which shows the current Quality Level and Certification Level of software

development process implemented in the Organization X. In addition, this presentation session was expected to give feedback about the implementation of the assessment conducted. The next section discusses about the result of the assessment as well as compares it with the previous case study result.

V RESULTS

As mentioned before, SPAC Model certifies software quality based on five factors that influence the software quality. The results of the case study are discussed further on the next subsections according to these factors. They are assessed from the perspective of completeness (how well the process was implemented and documented), consistency (how well the standard and procedures were followed) and accuracy (whether appropriate tools, method or technique were used)

1) The quality of process

The first assessed factor was the process. There are three types of process which were assessed: software development process, management process and support process. The achievements of these processes are discussed further.

- **Achievement for software development process**

Figure 1 shows the achievement for the software development process factor. The figure shows that the coding process is implemented efficiently. However, the requirement management process did not follow the standard and procedure as it only achieved 'Not satisfying' level. On the other hand, appropriate tools and technique have been used and appropriate documentation has been developed. The design phase also did not follow proper standard and procedure although it has been conducted following the proper practice and produce sufficient documentation. The least given attention is testing, although it is very important in determining the success of a project (Pressman, 2010). Overall, this activity only achieved 'Not satisfying' level.

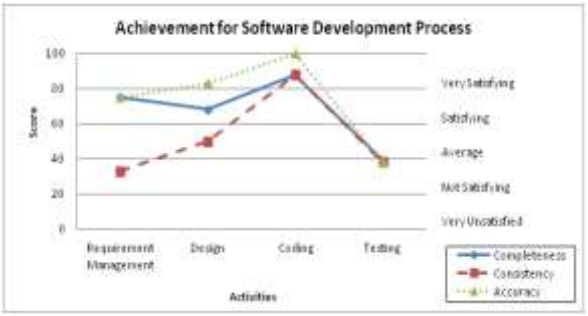

Figure 1. Achievement for software development process

- **Achievement for management process**

Figure 2 shows the achievement for the management process. The management processes were assessed from the perspective of project management, change management, quality management, technical review and risk management. Generally, all of these management processes were not implemented efficiently, whereby they achieved either 'Not satisfying' or 'Very unsatisfying'. This shows that AF System was implemented without proper planning. Organization X should give attention on this issue as efficient management is very important in order to produce high quality software (Sommerville, 2007; O'Regan, 2011).

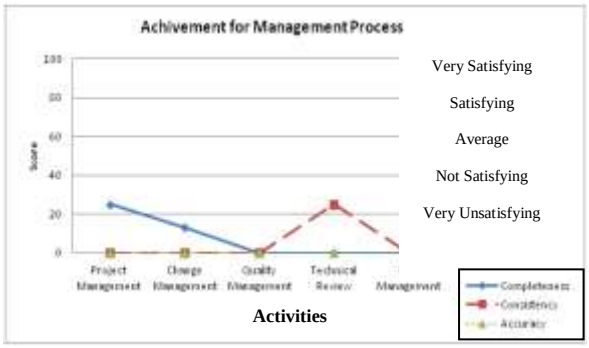

Figure 2. Achievements for management process

- **Achievement for support process**

The achievement for support processes were assessed based on the resource management, training, staff welfare and documentation. The outcome from these assessments is depicted in Figure 3. The resource management and staff welfare achieved 'Very satisfying'. However, the training for staff only achieved 'Average' level. This indicates that Organization X should provide more planned training for the staffs from time to time. Moreover, high attention should be given in producing the documentation as most of the activities were not documented well. Besides, the produced documentations were not updated as soon as new changes occur and never been verified by the management. However, documentation is very important in order to explain about the software systems and processes (Kajko-Mattson, 2008; Selic, 2009). In addition, according to Luqi, Lin, Berzins and Ying (2004), documentation is very important for the maintaining the system in future. Therefore, frequent updates and verification should be done from time to time. It will be very useful especially when there is a change in team members.

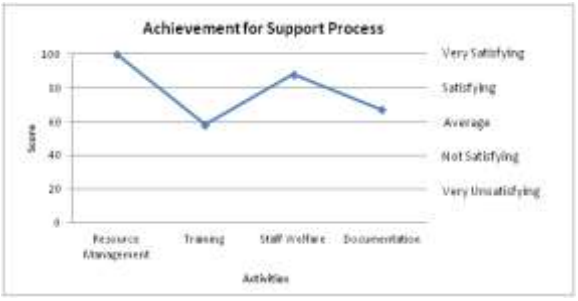

Figure 3. Achievements for Support Process

2) Quality of people involved

The second assessed factor is the quality of people involved in the software development, which are the developers. In addition, the involvement of management and customers also were taken into consideration.

- **Quality of developers**

Figure 4 shows the achievement of the developers of AF system. The qualities of the developers were assessed from the viewpoint of their interpersonal skills, management skills, technical skills, knowledge, experience and team commitment. Taken as a whole, all of the assessed attributes achieved 'Very satisfying' level, whereby the score achieved were 80 percent and above, except for the experience of the developers. This attribute achieved 'Satisfying' level. This shows that the developers of this system have good skills in developing software.

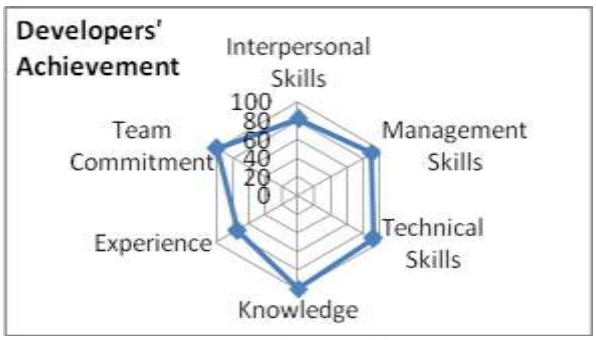

Figure 4. Developers' Achievement

- **Involvement of management and customer**

Outcome from the assessment showed that this organization emphasizes on the customer involvement during software development and customers also have given high commitment. However, the level of involvement by the management of this organization only achieved 'Average'. Thus, the management should increase their involvement during software development as involvement and support from management is considered important in order to produce high quality software.

3) Technology usage

The third assessed factor is the usage of technology. Among the assessed issues are the standard and procedure, tools and techniques and process origin. Result from the assessment shows that the usage of standard and procedure should be improved, as it is very essential in ensuring the uniformity of the development process implementation. This is because although the Organization X provides the standard and procedure that should be followed, however, the management does not inspect whether it is being followed properly or not. This attribute only achieved 'Vey unsatisfying' level. However, the usage of tools and techniques achieved 'Very satisfying', which demonstrates that Organization X emphasizes on this issue in order to ensure that the software were developed effectively and efficiently. Additionally, this system has been developed by using proper methodology, which is Rapid Application Development. This attribute achieved 'Very satisfying' level.

4) Project constraint

The fourth assessed factor is regarding the project constraint, by which the schedule and budget were the concern. The schedule of this project only achieved 'Average' level, as there is no stress given on the schedule of the project. Thus, the software development activities were conducted without having a proper planning. Yet, proper planning is vital in order to produce software which satisfies customer, which is completed on time, within budget and satisfies the user requirement (Sommervile, 2007; Nasution & Weistroffer, 2009). Also, as Organization X is the internal software developer, thus there is no budget included for the system.

5) Working environment

The final factor assessed is the working environment. It can be concluded that the facilities provided by Organization X is very conducive and comfortable for the developers. Thus, 'Very satisfying' level is achieved for the attribute comfort and security of working environment. The Cumulative Grade Point Average obtained from this assessment is 2.11, which indicates that the Certification Level achieved is Level II. The Quality Level attained for each attributes are represented in Table 1. From this result, it can be concluded that, generally the development of AF System did not follow proper software development practices. This is because most of the assessed practices were not implemented.

VI DISCUSSION AND CONCLUSION

Table1 shows the achievement for the Quality Level of each attributes for the two case studies. Generally, the achievement for both case studies either increases, decreases or remains in the same level. Among them, there are attributes which decreases drastically, from 'Very satisfying' to 'Very unsatisfying', for instance the level of consistency and correctness of project management, and the completeness and consistency of quality management. This shows that the best practices of software development are being neglected although they are important in producing high quality software. Nevertheless, the achievement for the support process increases from 'Satisfying' to 'Very satisfying'. Also, the quality of developers also has increased, particularly from the viewpoint of interpersonal skills, management skills and technical skills. This explains that with the increment of time, developers' experience grows wider and positively effects their skills and job quality.

In average, the Quality Level has decreased, thus the Certification Level for the current case study also decreased. The Certification Level achieved for the previous cased study was LEVEL IV, whereas in the current case study, only LEVEL II was achieved. This shows that when the time passes by, the best practices of software development are being disregarded. This issue need to be addressed by Organization X as the quality of produced software might decrease. With the decreasing of Certification Level, it reveals that continuous assessment on the software process is needed in order to know the Quality Level of software development process and to continuously improve the practices. Additionally, as SPAC Model provide the Quality Level for each attributes, thus the organizations can get guidance on which activities need to be improved further in order to produce high quality software.

Table 1: Assessment result of previous and recent study

Factors	Sub Factors	Attributes	Quality Level (2006)	Quality Level (2011)
Software Development	Requirement Management	Completeness	100 (VS)	75 (S)
		Consistency	58 (A)	33 (NS)
		Accuracy	75 (S)	75 (S)
	Prototype	Completeness	100 (VS)	75 (S)
	Design	Completeness	82 (VS)	68 (S)
		Consistency	88 (VS)	50 (A)
		Accuracy	83 (VS)	83 (VS)
	Coding	Completeness	83 (VS)	88 (VS)
		Consistency	88 (VS)	88 (VS)
		Accuracy	100 (VS)	100 (VS)
	Testing	Completeness	89 (VS)	39 (NS)
		Consistency	88 (VS)	38 (NS)
		Accuracy	50(A)	38 (NS)
Management Process	Project Management	Completeness	96 (VS)	25 (NS)
		Consistency	100 (VS)	0 (VU)
		Accuracy	100 (VS)	0 (VU)
	Change Management	Completeness	56(A)	13 (VU)
		Consistency	75 (S)	0 (VU)
		Accuracy	25 (NS)	0 (VU)
	Quality Management	Completeness	81 (VS)	0 (VU)
		Consistency	100 (VS)	0 (VU)
		Accuracy	75 (S)	0 (VU)
	Technical Review	Completeness	95 (VS)	0 (VU)
		Consistency	75 (S)	25 (NS)
		Accuracy	50(A)	0 (VU)
	Risk Management	Completeness	75 (S)	0 (VU)
		Consistency	75 (S)	0 (VU)
		Accuracy	50(A)	0 (VU)
Support	Resource Management	Completeness	75 (S)	100 (VS)
	Training	Completeness	100 (VS)	58 (A)
	Staff Welfare	Completeness	88 (VS)	88 (VS)
	Documentation	Completeness	67 (S)	67 (S)
Techno-logy	Standard & Procedure	Completeness	83 (VS)	29 (NS)
	Tools & Technique	Completeness	88 (VS)	100 (VS)
	Basic Process	Completeness	95 (VS)	85 (VS)
People	Developer	Interpersonal Skills	78 (S)	81 (VS)
		Management Skills	75 (S)	90 (VS)
		Technical Skills	79 (S)	92 (VS)
		Knowledge	88 (VS)	100 (VS)
		Experience	63 (S)	75 (S)
		Team Commitment	100 (VS)	100 (VS)
	Customer	Involvement	100 (VS)	100 (VS)
	Management	Involvement	100 (VS)	50 (A)
Constraint	Schedule	Accuracy	94 (VS)	50 (A)
	Budget	Accuracy	100 (VS)	100 (VS)

Environment	Working Environment	Comfort	100 (VS)	100 (VS)
		Security	100 (VS)	100 (VS)

Indicators:
VS: Very Satisfying NS: Not Satisfying
S : Satisfying VU: Very Unsatisfying
A : Average

As conclusion, this paper discussed about the implementation and the outcome obtained from the assessment conducted on AF System. This assessment was the second assessment conducted in Organization X, however, different software have been assessed. The assessments were based on the software development process approach. The first assessment was conducted in year 2006, while the second assessment was conducted in year 2011. Comparison between these assessments reveal that after five years' duration, the quality of software development process has decreased badly, from level IV to level II. These findings indicate that the software development best practices are being neglected. Therefore, it reveals that continuous certification process is vital in order to know the current status of software development process and to improve it. This is to ensure that the quality of produced software to be in high quality, based on the basic premise from Deming (1982), which is *'the quality of software product is influenced by the software process used to develop it'*.

REFERENCES

Cianfrani, C. A., Tsiakals, J.J. & West, J. (2009). *ISO 9001: 2008 Explained.* Milwaukee: American Society for Quality.

Cleland, G. L., Blanquart, J. P., Carranza, J. M., Froome, P. K. D., Jones, C. C. M. & Muller, J. F. (2003). A framework for the software aspects of the safety certification of a space system. *The Joint ESA-NASA Space-Flight Safety Conference.*

CMMI Product Team (2010). CMMI for Development V1.3 Technical Report.

Deming, W. (1982). Out of Crisis. *Cambridge, MA: MIT Center for Advanced Engineering Study.*

Fauziah Baharom (2008). A software certification model based on development process quality assessment. Unpublished doctoral dissertation, Universiti Kebangsaan Malaysia.

Fauziah Baharom, Jamaiah Yahya, Aziz Deraman & Abdul Razak Hamdan (2011). SPQF:software process quality factor for software process assessment and certification. *International Conference on Electrical Engineering and Informatics*, 1-7. doi: 10.1109/ICEEI.2011.6021526

Luqi, Lin, Z., Berzins, V. &Ying, Q. (2004). Documentation driven development for complex real-time systems, *IEEE Transactions on Software Engineering*, 30(12), 936-952. doi: 10.1109/TSE.2004.100

Nasution, M. F., & Weistroffer, H. R. (2009). Documentation in Systems Development: A Significant Criterion for Project Success. *Proceedings of the 42nd Hawaii International Conference on System Sciences*, 1-9. doi: 10.1109/HICSS.2009.167

Heck, P., Klabbers, M., Eekelen, M. (2010). A software product certification model. *Software Quality Journal*, 37-55. doi: 10.1007/s11219-009-9080-0

Jamaiah, Aziz & Abdul Razak (2007). A model and methodology for software product certification, *Proceedings of the National Conference Software Engineering and Computer System (Nacses07)*.

Kajko-Mattson, M. (2008). Problems in Agile Trenches. *Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM'08)*, 111-119. doi: 10.1145/1414004.1414025

O'Regan, G. (2002). *A Practical Approach to Software Quality.* Springer.

O'Regan, G. (2011). *Introduction to software process improvement*.Springer-Verlag London Limited.

Pikkarainen, M., & Mantyniemi, A. (2006). An approach for using CMMI in agile software development assessments: experiences from three case studies. *SPICE 2006 conference, Luxemburg*.

Pressman, R. S. (2010). *Software engineering a practitioner's approach 7th ed*. McGraw Hill.

Pyhajarvi, M., & Rautiainen, K. (2004). Integrating Testing and Implementation into Development, *Engineering Management Journal*, 16 (1): 33-39.

Rae, A., Robert, P., & Hausen, H.-L. (1995). *Software Evaluation for Certification Principles, Practice and Legal Liability.* England: McGraw-Hill.

Sedani, C. M., Lakhe, R. R. (2009). Critical factors to attain iso 9000 certification: a survey of Indian SMEs. *International Conference on Emerging Trends in Engineering and Technology (ICETET)*, 765-764.doi: 10.1109/ICETET.2009.65

Selic, B. (2009). Agile Documentation,Anyone?. *Software, IEEE*, 26(6), 11-12. doi: 10.1109/MS.2009.167

Sommerville, I. (2007). *Software Engineering 8th Ed.* Pearson Education Limited.

Vermesan, A. I. (1998). Software certification for industry-verification and validation issues in expert systems. *Proceedings of Database and Expert Systems Applications*, 3-14. doi: 10.1109/DEXA.1998.707373

Voas, J. (1998). The software quality certification triangle: crosstalk. *The Journal of Defense software engineering*, 11(11), 12-14.

Voas, J. (1999). User participation-based software certification. *EUROVAV '99 5th European Symposium on validation and verification of knowledge based systems*, 267-276

Wang, Y., Court, I., Ross, M., Staples, G., King, G. & Dorling, A. 1997. Quantitative Evaluation of the SPICE, CMM, ISO 9000 and BOOTSTRAP. *IEEE*. 57-6