SPAC: A Software Process Assessment and Certification Model

Fauziah Baharom[1], Aziz Deraman[2], Abdul Razak Hamdan[3]

[1]*Graduate Department of Information Technology*
College of Arts and Sciences
06010 Universiti Utara Malaysia, Sintok, Kedah, Malaysia
Tel : 04-9284701, Fax : 04-9284753
e-mail:[1]fauziah@uum.edu.my

[2]*Department of Computer Science*
[3]*Department of Science and Information Systems*
Faculty of Information Science and Technology
Universiti Kebangsaan Malaysia
43600 Bangi, Selangor
E-mail: [2]a.d@pkrisc.cc.ukm.my, [3]arh@ftsm.ukm.my

ABSTRACT

This paper discusses the development of software process assessment and certification (SPAC) model as an effort to ensure and improve the quality of software process. To ensure the compatibility with current practice, the SPAC model was designed by referring to existing models and standards namely ISO 9000, ISO/IEC 15504, Bootstrap and Capability Maturity Model (CMM). The main thrust of the model development is based on five main factors of software quality: process, people, development technology, working environment and project conditions. The distinguish features of the SPAC model are the assessment form, the assessment and certification procedures and the mechanisms for determining the quality and certification levels. In addition, the model also put emphasis on the implementation of team based assessment and the use of multiple data gathering techniques to improve the quality of the data collected.

Keywords:

SPAC Model, Software Process Assessment, Software Certification, Software Quality

1. INTRODUCTION

Our society nowadays is highly dependent on computer and software in managing their daily life. As a result, customers are not only interested in acquiring software faster but also emphasizes on quality of the software. Therefore, producing high quality software and on time delivery are certainly important. In order to understand on the current practices of software development process in Malaysia, we had conducted an empirical study where 44 organizations had responded to the study (Baharom et al. 2005). Findings from the survey found that software organizations in Malaysia were lacking in implementing good software development practices. Out of 44 respondents, the results show that 30% were not following any standards and 31.8% were developing software without follow any methodology. All respondent indicates that their software projects were facing with quality problems such as failed to finish on time (56.8%), exceed budget (25%), not meet expected quality level (27.3%) and need to fix problem after deployment (77.3%). The analysis also showed that majority of the organizations had no dedicated test professional and formal testing was performed by software developers themselves. 90.9% of the respondents agreed that software testing by itself is not sufficient to establish confidence that software is fits for its intended use. More over, the study also indicates that an independent certification may be required in assuring the quality of software.

In general, the ultimate goal of the software certification is to assess the quality of software product. However, the quality of software is difficult to determine without implementing it for a long period (Sommerville, 2001). Based on a basis premise that "*the quality of product is largely governed by the quality of process used to develop its*" (Deming 1982; Humphrey 1989), quality can be assured by looking at how carefully it was built. Therefore, many studies have been conducted in the area of software process certification. Unfortunately; these studies are not providing complete practical mechanisms for assessing and certifying software process quality.

Thus, this study focuses on the development of software process assessment and certification (SPAC) model. The primary goal of the model is to provide a complete guideline for assessing and certifying software process in order to ensure that the software product has been developed effectively and efficiently. Furthermore, through development of the SPAC model, software customers can choose the software that meets their expected quality levels and also can help organizations to plan for continuous software processes improvement.

This paper aims to introduce the SPAC model. The paper is structured as follows. Section 2 discusses on the need for software process assessment and certification, while Section 3 describes about the goals and objectives of the SPAC model. Section 4 gives an overview of the SPAC model, and finally Section concludes the paper.

2. THE NEEDS FOR SOFTWARE QUALITY AND CERTIFICATION

IEEE defines quality in the software context as the features or characteristics used to assess the quality of a system or component (IEEE 1993). Sanders and Curran (1994) believes that quality is essential for competitiveness, survival and success. These are the reasons why the software industries require mechanism for software's quality assurance.

Certificate is always been seen as a sign of quality and software certification is a process or formal procedure by which a third party gives written assurance that a product, process or service conforms to the specified characteristics (Rae et al., 1995; Cleland et al., 2003). Software certification is essential because:

i. organizations will emphasize on the use of standards and procedures (Tripp 2002)

ii. users' confidence on the quality of software will be increased (Stafford & Wallnau, 2001)

iii. it is an alternative approach to monitor and encourage on continuous improvement of software process quality (Deraman et a., 2007)

iv. increased ability to compete in the market (Vermesan, 1998)

3. THE SPAC MODEL

As discussed in Section 1, the primary goal of the SPAC model is to assure that the software development process has been carried out carefully in order to meet the expected quality criteria, be delivered on time and within budget.

The SPAC model has been constructed by integrating the requirements from several existing standards and models of software process assessment, and these are:

- ISO/IEC 15504 (O'Regan, 2002; SPICE, 1995) which is the emerging international standard for software process assessment,
- Bootstrap (Kuvaja, 1999) which is a software process assessment and improvement methodology for the European software industry,
- ISO 9001 and ISO 90003 (O'Regan, 2002) which are the international standards that are commonly used for certifying quality management system of an organization
- Capability Maturity Model (CMM) (Paulk et al., 1997) is a model or framework for determining the maturity levels of software organization.

The main elements of the SPAC model are the structure of the assessment form, the assessment and certification procedure and the mechanisms for determining quality and certification level.

3.1. The Assessment Form

The SPAC model was designed not only on assessing the technical aspect but it also emphasizes the managerial, human and environmental aspects. It has an assessment form that can be used as the instrument data gathering. The assessment form consists of identified metrics which have been organized under five main factors: processes, people, development technologies, working environments and project conditions. The design of assessment form follows the outline of the SPAC quality factors as shown in Figure 1. Each metric comprises of a set of measures, which are used as questions for the data gathering. Each individual measure is assessed by using the "five-likert scale" format.

3.2. The Assessment Approach

The assessment process are normally conducted either through first, second or third party assessment. However, the SPAC model recommends that every assessment to be conducted by using the collaborative approach (Byrnes & Philips, 1996). It is a team-based approach, where members of the organization (such as project leader/manager and developer) must be part of the assessment team. To be effective, the assessment team should consist of at least two members, but not more than eight members (Humphrey, 1989). The team members are gathered from various sources, which includes software engineering experts or/and software assessment experts (SPICE, 1995). The reasons for these recommendations are to get more rigorous and reliable results and to make the process easier and faster. In addition, the assessment data is collected by using multiple data gathering techniques proposed by (SPICE, 1995) such as documents review, interview and observation to improve the quality of the data collected.

3.3. The Assessment and Certification Process

The model consists of three main phases of implementation as adapted from the BOOTSTRAP technique (Kuvaja, 1999), which namely as the Preparation Phase, the Execution Phase and the Post Assessment Phase. These phases are then composed into set of activities; 16 activities all together. These activities fundamentally intend to provide guidance to facilitate the whole process of forming assessment team, collecting data, analysis data, reporting and presenting results. However, the process, steps should not be taken as a cookbook but rather it should be adapted conveniently to meet the requirements of a particular situation. It can be considered as mechanical guidance of software certification process.

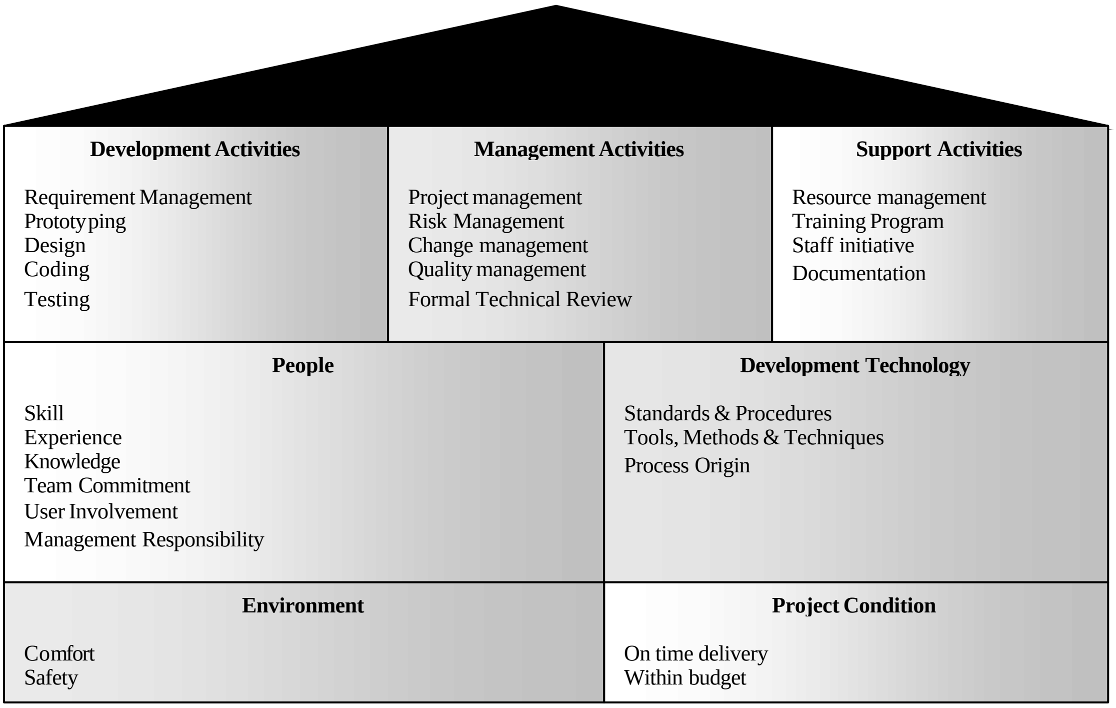

Figure 1. The SPAC Quality Factors

3.4. The Quality and Certification Determination

The assessment processes and results are the basis of certification process. Therefore, achieving reliable results is certainly important. Reliability can be obtained by applying precise scoring rules and rating scales. During the assessment process, the likert-scale value that represents answer of each measure is converted into appropriate integer values between 0 and 4. Then, the total score for each quality attribute or metric is calculated. Based on the total score obtained, the quality level of each metric is determined on a satisfaction scale from 0% to 100% as shown in Table 2. This study propose 5 quality levels which are either "not fully satisfied", "not satisfied", "partially satisfied", "largely satisfied" and "fully satisfied". The rating scale has been edited from the ISO 15504 (or SPICE) ratings of capability level of process attributes (O'Regan, 2002).

The final process is determining the certification level of the software process. The SPAC model proposed five certification levels, which are incomplete, poor, moderate, good and excellent. The certification level is determined by using the value of Total Score Average (TSA), which is adopted from the Grade Point Average (GPA) concept, normally used to determine student's performance. The "incomplete" level will be assigned to the development processes that are carried out without following any software development methodology. Table 3 presents the description of each certification level.

4. CONCLUSION

The SPAC model was developed with the aim to assess and certify the quality of the software development process. The model has been constructed based on existing standards and models and these are ISO 9000, ISO 15504, CMM and Bootstrap. The SPAC model focuses on five main factors of software quality: the quality of software process used to develop the software, the quality of people involved, the used of development technology, working environment and project conditions in term of budget and scheduling. The main elements of the SPAC model include the assessment form, assessment and certification procedures and mechanisms to determine quality and certification level. To be more effective, eliminating biasness and obtaining rigorous results, the model is recommended to be used in a team-based assessment through a collaborative approach. Information from the implementing the model can benefit to both, the developers and customers. Developers can plan software development practices improvement, while customers are confident on the quality of software. As a conclusion, the development and construction of the SPAC model can be judged as having achieved its expected goals and objectives.

REFERENCES

Byrnes, P. and Phillips, M., Software Capability Evaluation, (1996). (http://www.sei.cmu.edu/publications/documents/96.reports/96.tr.002.html)

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. *IEEE.*

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

Deraman, A., Yahya, J. H., Baharom, F., Fadzlah, A. F. A.& Hamdan, A. R. 2007. Continuous quality improvement in software certification environment. *Proceeding of the International Conference on Electrical Engineering and Informatics (ICEEI 2007)*. 11-17.

Fauziah Baharom, Aziz Deraman, Abdul Razak Hamdan. (2005). A survey on the current practices of software development process in Malaysia. *Journal of Information and Communication Technology (Journal of ICT) Volume 4*

Humphrey, W. S. (1989). *Managing the software process.* Addison-Wesley

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

Kuvaja, P. (1999). BOOTSTRAP 3.0 - A SPICE Conformant Software Process Assessment Methodology, *Software Quality Journal* **8**: 7-19

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

Paulk, M. C., Curtis, B., Chrissis, M. B., and Weber, C. V., (1997). The Capability Maturity Model for Software, *IEEE Computer Society Press*: 427-438

Rae, A., Robert, P., and Hausen, H.-L., (1995). *Software Evaluation for Certification,* McGraw Hill

Sanders, J., and Curran, E., (1994). *Software Quality: A framework for success in software development and support*, Addion-Wesley

Software Process Improvement and Capability dEtermination (SPICE), (1995). (http://www.sqi.gu.edu.au/spice/)

Sommerville, I. 2001. *Software Engineering*. 6th. Addison Wesley.

Stafford, J.& Wallnau, K. 2001. Is third party certification necessary? *Proceedings of the 4th ICSE Workshop on Component-based Software Engineering: Component Certification and System Prediction.* 13-17.

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

Vermesan, A. I., (1998). Software Certification for Industry - Verification and Validation Issues in Expert Systems, *Proceedings of the 9th International Workshop on Database and Expert Systems Applications*