# ESPAC MODEL: EXTENDED SOFTWARE PROCESS ASSESSMENT AND CERTIFICATION MODEL FOR AGILE AND SECURE SOFTWARE PROCESSES

Shafinah Farvin Packeer Mohamed[1], Fauziah Baharom[1], Aziz Deraman[2]
[1]School of Computing, Universiti Utara Malaysia, Sintok, Kedah, Malaysia
[2]Faculty of Science and Technology, Universiti Malaysia Terengganu, Kuala Terengganu, Terengganu, Malaysia
E-Mail: shafinah@uum.edu.my

## ABSTRACT

In today's business environment, agile and secure software processes are essential since they bring high quality and secured software to market faster and more cost-effectively. Unfortunately, some software practitioners are not following the proper practices of both processes when developing software. There exist various studies which assess the quality of software process; nevertheless, their focus is on the conventional software process. Furthermore, they do not consider weight values in the assessment although each evaluation criterion might have different importance on the project. Consequently, software certification is needed to give conformance on the quality of agile and secure software processes. Therefore, the objective of this paper is to propose ESPAC (Extended Software Process Assessment and Certification) Model which addresses both software processes and considers the weight values during the assessment. The model has been constructed by using deductive approach, whereby the theoretical and exploratory studies were conducted in order to determine the components of the proposed model. The ESPAC Model consists of six components, which are the target, evaluation criteria, reference standard, data gathering technique, assessment phases and synthesis technique. The proposed model has been validated by seven software practitioners from the practical perspectives through focus group discussion. The validation result shows that the ESPAC Model gained their satisfaction and able to be practically executed in the real environment.

**Keywords:** software process assessment and certification, agile and secure software processes, analytic hierarchical process.

## INTRODUCTION

Software is among the most widely used product in human history, nevertheless its failure rate is one of the highest among any product in human history (Jones & Bonsignour, 2012). On the other side, software practitioners always claim that they produce high quality software, yet there still exist customer complains on the quality of software produced to them (Weber-Jahnk, 2011). Therefore, it is vital for the software practitioners to implement best practices of agile and secure software processes to improve the quality of software process that they implement in order to produce high quality software within time and budget in today's business environment. Nevertheless, observations from previous studies show that the software practitioners are left far behind the theoretical best practices. This was observed by Brooks back in the year 1987 (Cater-Steel, 2004; McConnell, 2002), and still can be seen today (Fauziah, Aziz, & Abdul Razak, 2005; McConnell, 2002).

Consequently, certification becomes a mechanism to give conformance on the quality of software (Heck, Klabber, & Eekelen, 2010; Aziz, Jamaiah, Fauziah, Amalina Farhi, & Abdul Razak, 2007). Certification is defined as "the procedure by which a third party gives written assurance that a product, process or service conforms to a specified characteristics" (Rae, Robert, & Hausen, 1995). With certification, customers feel more confident on the quality and dependability in selecting the desired organization for investments. Moreover, certification involves with assessments, thus it is possible to reduce the risks of software failure.

Voas (1998) summarized that certification in the software industry can be implemented in three approaches which are personnel, product and process. Even though many researchers believe product based approach can give confidence to customers about the quality of software (Heck et al., 2010; Jamaiah, 2007; Voas, 1999), at the same time, they admit that the quality assessment for product based approach is hard to be practiced without implementing the software for a certain period of time. Thus, based on the Deming's premise that "the quality of product is influenced by the quality of process used to develop it" (Deming, 1982), it is believed that process based software certification can be an alternative solution.

Numerous studies were intended to produce models and standards for software process improvement (SPI) including ISO/IEC 15504 (O'Regan, 2002) and Capability Maturity Model Integration (CMMI Product Team, 2010). On the other hand, the ISO 9000 (Sedani & Lakhe, 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, Jamaiah, Aziz and Abdul Razak (2011) mainly focuses on certifying software process in order to ensure that the software process was carried out effectively and efficiently. Unfortunately, this model did not address the agile and secure software processes in its assessment. However, in today's business environment, both approaches have become as determinant factors to produce high quality software (Merkow & Raghavan, 2010; Pressman, 2010). Furthermore, the existing software process certification models and standards do not consider weight values in their assessment even though the assessment involves multi criteria. On the other hand, weight value allocation is very important to be considered especially when the assessment process involves multi criteria.

Consequently, a research was conducted to construct Extended Software Process Assessment and Certification (ESPAC) Model which addresses these software processes, besides considers weight values in the assessment. The objective of this paper is to discuss the construction of ESPAC Model by utilizing the outcomes from the theoretical and exploratory studies. Furthermore, it discusses the results from the validation performed on the proposed model through focus group discussion.

This paper is started with the issues in existing software process certification models and standards, continued with the research approach. Then, the ESPAC Model is introduced followed by the validation of ESPAC Model. The paper is ended with the conclusion.

## ISSUES IN THE EXISTING SOFTWARE PROCESS CERTIFICATION MODELS & STANDARDS

There exist many studies which have been conducted in the field of process based software certification, but most studies were intended to produce models and standards for software process improvement (SPI) including ISO/IEC 15504 and Capability Maturity Model Integrated (CMMI) (O'Regan, 2002). On the other hand, the ISO 9000 (Sedani & Lakhe, 2009) provides a mechanism to certify the quality system of an organization. The organization needs to go through an evaluation which is run by an independent party to assure that the quality practices performed by the organization are complied with the standard. However, ISO 9000 focuses on the quality management, not specifically on the software process.

To fill in this gap, Fauziah et al. (2011) have introduced the Software Process Assessment and Certification (SPAC) Model which mainly focuses on certifying software process in order to ensure that the software process was carried out effectively and efficiently. SPAC Model considers five factors that influence the quality of software process as the reference model, which are the process performed, the quality of people involved, the working environment, the use of development technology and project condition, which focuses on the conventional software process. Conventional software process refers to "specification-based software development, which is based on completely specifying the requirements up front then designing, building and testing the system with emphasize given on documentation rather than the software itself" (Sommerville, 2007).

The Capability Maturity Model Integrated (CMMI) (SCAMPI Upgrade Team, 2011) on the other hand, includes project management, process management, engineering and support. CMMI has included guidelines and notes for software practitioners who implement CMMI in the agile environment (CMMI Product Team, 2010), however, they are only general guidelines and included only for certain process area. As defined by Boehm & Turner (2005), agile is "lightweight software development approach which emphasizes on iterative, incremental, self-organizing and emergent practices".

Moreover, the existing security standard such as ISO/IEC 27001 tends to focus on information security management system and only focuses generally on secure software process. ISO/IEC 21827 (Davis, 2009) conversely concentrates on security engineering practices and focuses on the maturity of system security management. According to McGraw (2004), secure software process is "about building secure software: designing software to be secure, making sure that software is secure, and educating software developers, architects, and users about how to build secure things".

It should be noted that limited studies are available on agile and secure software processes in the existing software process certification models and standards eventhough both approaches have become as determinant factors to produce high quality software in today's business environment (Merkow & Raghavan, 2010; Pressman, 2010). Therefore, this has motivated the present research. Moreover, these two software processes have been partially addressed by the two most influential software process models and standards providers, which are Software Engineering Institute (SEI) through CMMI (CMMI Product Team, 2010) and International Organization for Standardization (ISO) through standards which emphasize on security issue: ISO/IEC 27001 (Evans, Tsohou, Tryfonas & Morgan, 2010) and ISO/IEC 21827 (Davis, 2009). This shows that both software processes are important to be addressed in the software process certification model for the current business environment.

Additionally, the existing software process certification models needs further improvement on the decision making technique to get better quality and consistency on the certification decision made. This is because the software certification process involves with assessment based on multi criteria/factors, thus each of them must be weighted since they might have different importance (Malczewski, 1999; Yoon & Hwang, 1995). Weight can be defined as "a value assigned to an evaluation criterion that indicates its importance relative to other criteria under consideration" (Malczewski, 1999). However, despite of its importance, little attention has been directed to consider weight values for the assessed criteria.

Since software process assessment involves with multiple criteria, hence the Multi Criteria Decision Making (MCDM) technique is an appropriate technique for determining the weight for each assessed criteria. MCDM refers to "making preference decision over the available alternatives that are characterized by multiple, usually conflicting attributes" (Triantaphylluo, 2000). There are many techniques to determine the weight in

MCDM, however, AHP is the most widely used technique (Vaidya & Kumar, 2006). It has been successfully implemented in the domain of evaluation (Al-Tarawneh, 2014; Zhou & Liang, 2013) On top of that, AHP includes the consistency checking in the judgment, which is essential in order to ensure that the judgments have been made consistently (Liberatore & Nydick, 1997). Consequently, these have motivated the research to improve the assessment technique in the software certification process by incorporating weight values through the adaptation of AHP (Saaty, 1990).

Based on the initiative made by SPAC Model, this research has overcome the above mentioned issues by enhancing the software process certification model. With the enhancement made, the certification process can be conducted in a broader aspect since it includes agile and secure software processes. Furthermore, with the weight values allocation, the proposed model produces more consistent decision on the certification result.

## RESEARCH APPROACH

To achieve the objective of this research, deductive approach, which is also known as 'top-down' approach, was performed (Trochim, 2006). Deductive approach begins from general ideas and ends with specific conclusions, whereby the conclusions are made based on a known general premise or something known to be true. There are four phases in conducting the research which are theoretical study, exploratory study, ESPAC Model development and ESPAC Model evaluation. Table-1 shows the activities and outcomes of each phases.

**Table-1.** The phases, activities and outcomes.

| Phases | Activities | Outcomes |
|---|---|---|
| **Theoretical study** | • Identify problems in software certification<br>• Identify the factors, evaluation criteria and software process that influence the quality of agile and secure software processes<br>• Review the techniques for determining weight | • Problems and generic features of software certification<br>• List of factors, evaluation criteria, and agile and secure software processes<br>• List of techniques for determining weight |
| **Exploratory study** | • Instrument design<br>• Identify the sample<br>• Instrument validation & pilot study<br>• Data collection<br>• Data analysis | • Instrument<br>• Pilot test report<br>• Set of agile and secure software processes<br>• Improved reference standard |
| **ESPAC model development** | • Determine the components<br>• Build the reference standard by including the evaluation criteria and agile and secure software processes<br>• Improve the decision making technique by using AHP and WSM<br>• Determine the data gathering technique<br>• Determine the assessment phases | Proposed software process certification model which focuses on the agile and secure software process and considers weight values during software process assessment |

| Phases | Activities | Outcomes |
|---|---|---|
| **ESPAC model evaluation** | • Verification: Expert review<br>- Identify experts<br>- Determine verification criteria<br>- Collect & analyze feedbacks<br>• Validation: Focus group<br>- Identify experts<br>- Determine validation criteria<br>- Collect and analyze feedbacks | • Feedbacks from verification performed<br>• Improved ESPAC Model<br>• Feedbacks from validation performed |

## THE ESPAC MODEL

Figure-1 illustrates the ESPAC Model, which is an extended software process assessment and certification model that focuses on the agile and secure software processes. It is aimed for assessing and certifying the quality of agile and secure software processes. At the end of the certification exercise, the model produces certification level and quality levels of the assessed software processes. The model is formulated by referring to existing software process certification models or standards which are SPAC Model (Fauziah et al., 2011), Capability Maturity Model Integrated (CMMI Product Team, 2010), ISO/IEC 15504 (O'Regan, 2002), ISO/IEC 27001 (Evans et al., 2010) and ISO/IEC 21827 (Davis, 2009). Besides these, the agile principles and methods were referred for eliciting the agile software process (Agile Manifesto, 2001). For eliciting the secure software process, three most prominent models were referred, which are the Microsoft SDL, McGraw's Touchpoints and CLASP (De Win, Scandariato, Buyens, Gregoire & Joosen, 2009). Additionally, the decision making technique is improved by incorporating AHP (Saaty, 1990) for weight allocation in the the synthesis technique.
The key activity in software certification is evaluation. Thus, the Evaluation Theory (Scriven, 1991) is very closely related to the certification process. Therefore towards constructing the proposed model, the components were determined based on this theory. The components are the target, evaluation criteria, reference standard, data gathering technique, synthesis technique and assessment process. Each components is elaborated further in the next sub sections. In a nutshell, the ESPAC Model is developed based on the outcomes from the theoretical study as well as findings from exploratory study. Interested readers are directed to the findings from the exploratory study which can be attained from Shafinah, Fauziah and Aziz (2014a).

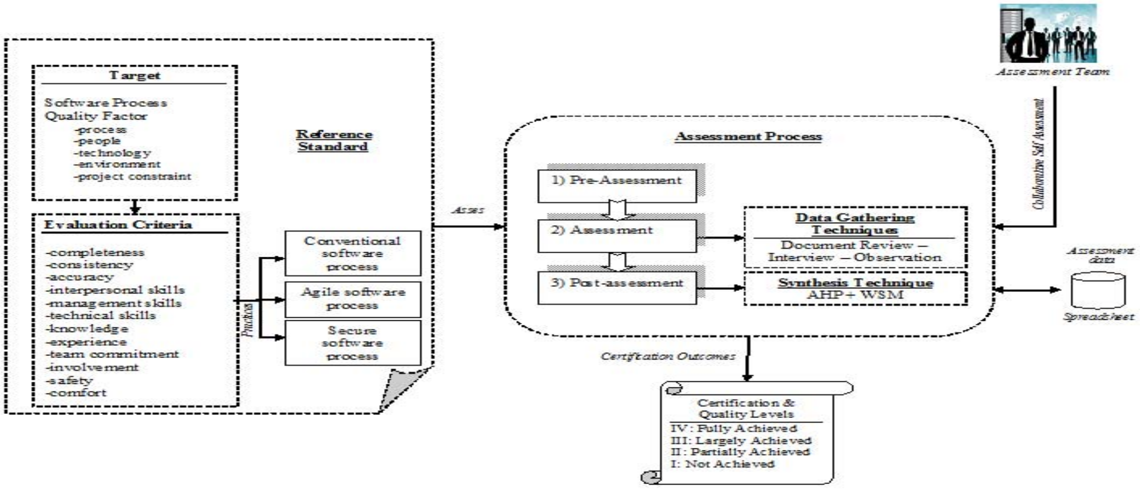

**Figure-1.** The ESPAC model.

### The target

The target is 'the object under evaluation'. Defining the target is the first essential process in any assessment. By defining the target, the assessor can get insight on what should be assessed. In this research, the target is software process. Software process is defined as "set of activities undertaken to manage, develop and maintain software systems in order to produce a software system, executed by a group of people organized according to a given organizational structure and counting on the support of techno-conceptual tools" (Acuna, Antonio, Ferre, Lopez & Mate, 2000). Nevertheless, since software process is performed by human, therefore there are other factors which can influence the quality of software. They are the people, technology used, project constraint and environment (Fauziah et al., 2011; Hazzan and Dubinsky, 2009; Ares, Garcia, Juristo, Lopez & Moreno, 2000). Each of these factors is decomposed to sub factors. They are represented in a hierarchy tree, as depicted in Figure-2.

### The evaluation criteria

The evaluation criteria are 'the characteristics of the target'. Basically the evaluation criteria are comprised of the characteristics that need to be accomplished in order to achieve the effectiveness and efficiency of software process. The effectiveness is measured based on the completeness, consistency and accuracy of the process in developing software which can fulfill customers' expectations through involvement of good quality people, use of appropriate technology and stability of working environment. On the other hand, the efficiency is measured based on the capability of software process to produce software within estimated time and budget (Fauziah et al., 2011). Each of the factors is assessed based on particular criterion, which is represented by the lowest level of the hierarchy tree in Figure-2.

### The reference standard

Based on the defined target and evaluation criteria, the reference standard is constructed. It consists of the best practices of agile and secure software processes. The Quality Function Deployment (QFD) (Zultner, 1992) approach is utilized to organize them. Each evaluation criteria is assigned with appropriate agile and secure software software processes. They are obtained through theoretical study and exploratory study conducted in this research.

### The data gathering technique

The data are gathered by using multiple techniques, which are assignation techniques and opinion (Ares et al., 2000). The assignation techniques used in this model are the document review and interview. On the other hand, the opinion technique denotes the observation. Using multiple data gathering technique can improve the understanding for the assessment team and give better confirmation on the assessment made (SCAMPI, 2011). Table-2 depicts the data gathering techniques used for each assessed factor.

**Table-2.** The data gathering technique.

| Factors | Data gathering techniques |
|---|---|
| Process | Document review +Interview (for clarification) |
| People | Interview |
| Technology | Document Review + Interview |
| Project constraint | Document review |
| Environment | Observation |

### The synthesis technique

Synthesis technique is "the technique used to judge each criterion, and in general, to judge the target, obtaining the results of the evaluation". In this research, there are two main stages for synthesizing. First stage is to determine the weight for each evaluation criterion, which is accomplished by performing the AHP technique (Saaty, 1990). The second stage is to perform the assessment by comparing the reference standard with the practices implemented during software development. Each practices is assigned with appropriate score which ranges from 1 (Never) to 5 (Always). Then, the total scores are obtained for each evaluation criterion by utilizing the Weighted Sum Method (Mollaghasemi, 1997).

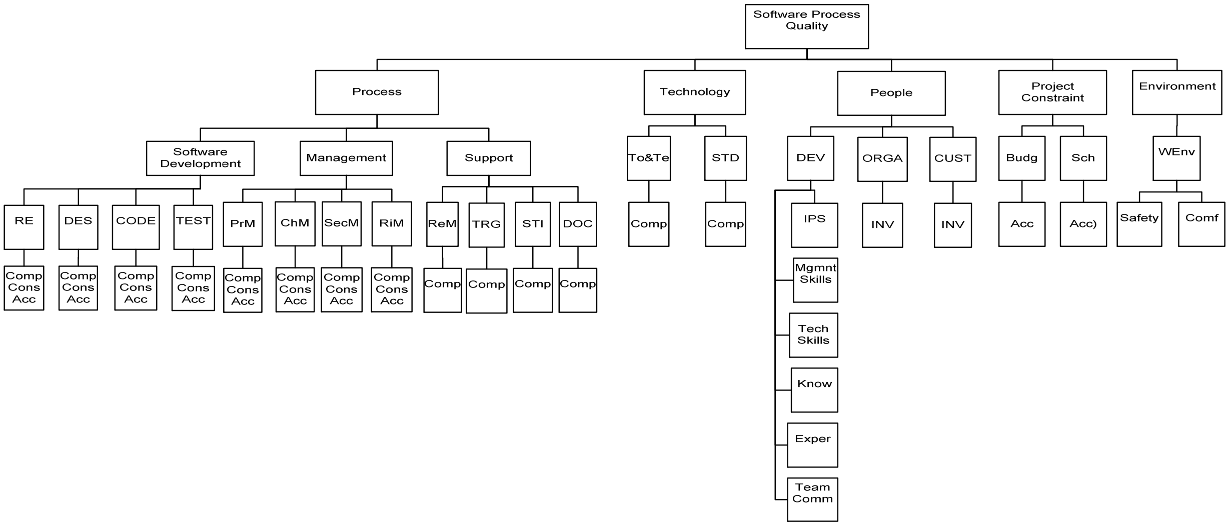

**Figure-2.** The target and evaluation criteria.

At the end of the assessment, ESPAC Model produces two certification outcomes, which are the quality levels of each evaluation criterion and the certification level for the whole assessment. Both of them are based on the defined Achivement Index, as depicted in Table-3. This Achievement Index is adapted from ISO/IEC 15504. The highest quality level is 'Fully achieved', while the lowest quality level is 'Not Achieved'.

**Table-3.** The achievement index.

| Score Values | Descriptions |
|---|---|
| **Level IV**<br>Fully Achieved<br>86 ≤ Score ≤100 | This level indicates a fully satisfying achievement. The software processes were implemented effectively, systematically and perfectly or almost perfectly. |
| **Level III**<br>Largely Achieved<br>51≤ Score ≤ 85 | This level indicates a largely satisfying achievement. The software processes were implemented quite systematically. However, some software processes of low performance exist. |
| **Level II**<br>Partially Achieved<br>16 ≤ Score ≤ 50 | This level indicates a partially satisfying achievement. A systematic approach has been used; however almost all of the asessed software processes were not implemented properly. |
| **Level I**<br>Not Achieved<br>0 ≤ Score ≤ 15 | This level indicates unsatisfying level of achievement. The software processes were not implemented systematically and below average. The methodology usage was neglected. The software process is considered as fail to achieve its goal. |

**The assessment process**

There are three phases of assessment process. Each of the phases has several activities, as adapted from SCAMPI (SCAMPI, 2011), SPAC Model (Fauziah et al., 2011) and Lascelles and Peacock (1996):

i. Pre-assessment: develop commitment, identify and analyze the candidate project, plan the assessment, form the assessment team, prepare the assessment team and prepare for assessment conduct.

ii. Assessment: prepare assessment participants, perform JAD session, review documents, perform interviews, observe, record the information gathered and synthesize and analyze data.

iii. Post assessment: determine certification level and quality levels, present assessment result and gather feedback, collect lessons learned and prepare technical report.

The assessment process applied in ESPAC Model is proposed as collaborative self-assessment method. This method is adapted from self-assessment (Serkani, Mardi, Najafi, Jahanian & Heart, 2013; Tari & Heras-Saizarbitoria, 2012; Lascelles & Peacock, 1996) and collaborative assessment (Fauziah et al., 2011; Jamaiah, 2007). By applying collaborative self-assessment, the assessment team consists of organization's own people. It is lead by a project manager and composed of assessors from outside of the team being assessed (software practitioners from other software development team in the organization). This is aimed to eliminate biased assessment (Fauziah et al., 2011; Jamaiah, 2007;Fabbrini et al., 2006; Voas, 1999). Furthermore, the cost can be reduced since the assessment is only performed within organization (Ritchie & Dale, 2000).

Additionally, one representative from the assessed team co-operates as one of the assessment team members to facilitate ideas exchange between the assessment team (Lascelles & Peacock, 1996). Moreover, the assessment process can be accelerated since the project is already understood by the representative (Voas, 1999; Vermesan, 1998). The assessors are among software practitioners who have knowledge in software engineering and assessment.

## VALIDATION OF ESPAC

As discussed earlier, ESPAC Model is an extended software process assessment and certification model which focuses on the agile and secure software processes. In order to ensure the satisfaction and practicality of the proposed model, it was validated by software practitioners through focus group discussion (Shafinah, Fauziah & Aziz, 2014b; Kontio, Bragge & Lehlota, 2008; Krueger, 1994). There were seven software practitioners who attended the session. They were the team leaders, Scrum Master, Application Lifecycle Program Manager, architect and programmers. They also had experience in agile and secure software processes.

During the focus group session, the software practitioners were briefed about ESPAC Model. Then, each of them chose one of the projects that they have completed and assessed the software processes that they implemented by applying the ESPAC Model. After completing the focus group session with the participants, the researcher analyzed the data obtained from the focus group. The total scores for the assessment and certification exercise were calculated. Then, the quality levels as well as the certification level for each projects were obtained. The outcomes were then reported in technical reports by representing them in tables and charts. These technical reports were then emailed to the participants. Based on the report, the participants emailed back their feedbacks on the validation of the ESPAC Model.

The software practitioners validated the ESPAC Model based on a predefined set of criteria, which are adapted from previous studies (Al Tarawneh, 2014; Kunda, 2003; Kitchenham & Pickard, 1998). These criteria are gain satisfaction, interface satisfaction and task support satisfaction. Gain satisfaction measures whether the model is beneficial. Interface satisfaction on the other hand, focuses on the characteristics of the interface in terms of presentation, format and processing efficiency. Furthermore, task support satisfaction measures whether the model achieves its intended objectives and satisfies evaluators. The subsequent subsection discusses the outcomes from one of the assessed and certified projects. First the background of the project is provided, continued with the results for each assessed factors. Then, the certification results are provided followed by the feedbacks gained from the validation.

### Project background

The assessed project is a portal developed for an organization. This project was developed by one of the Multimedia Super Corridor (MSC) status organizations in Malaysia. The project started on the 10 September 2011 and ended on the 25 December 2012. There were six main functions in this system, which are place order, view order, view products, card activation, order history and manage profile. This software was developed by using multiple languages such as Java, Struts 2 and Jquery, while Informix was utilized to keep the data. The method used during the software development was Scrum. It was developed by 10 to 20 team members.

### The process

Basically the project was developed by following the best practices of agile and secure software processes. However, several software processes were neglected, especially regarding the support process. The top management gave less emphasis to the staffs' welfare. They did not ensure that the staffs worked for 40 hours or less per week as suggested by agile (Salo & Abrahmamsson, 2008). On top of that, the resources needed for the project were only sometimes available for the needed periods. This can become as one of the impediments for the project. Also, the top management gave less attention for management trainings; nevertheless, technical trainings were given emphasis.

In addition, although this software involves with transactions which require high security features, unfortunately the security requirements were not documented. They were discussed from early stages but left to the implementation team as it is assumed that they will deal with security. Additionally, some of the software design practices might be neglected such as performing external review. This practice is important because errors are easier to be spotted by someone who did not develop the code. Moreover, the countermeasures for the identified threats were not designed and documented. This might cause the team to overlook the possible threats.

### The people

The software practitioners, customers and organization are assessed for the people factor. The software practitioners are highly knowledgeable, experienced, gave commitment to the project and are good in technical skills. However, they lacked interpersonal and management skills.

Additionally, the customers' commitment is very important for the team working in agile environment. Basically the customers of this project were involved throughout the software development and know the business domain. However, they were sometimes not able to collaborate with the team members and had problems in making decisions.

Furthermore, the organization's involvement is very crucial in agile environment, as support from the top management has high influence on the success of agile implementation. This organization ensures that agile is universally accepted, which is essential. However, it still practices hierarchical organizational culture instead of cooperative culture as enforced by agile.

### The project constraint

The project was planned and managed accurately, as it was finished within the estimated budget and time. However, the team might neglect some of estimation practices such as performing estimation continuously for solution in-hand.

### The technology

The achievement shows that the organization emphasizes on the use of tools and technology in order to implement the software process effectively and efficiently. Nevertheless, although the use of standard and procedure is important for ensuring the uniformity of the software process and produced work product, less emphasis was given by the top management to monitor its implementation among the staffs.

### The working environment

The organization emphasizes on the safety and comfort of the working environment. However, the appropriate hardware and software should be provided to backup the data.

### Certification results

Table-4 depicts the quality levels for the evaluation criteria, whereby majority of them attained Level III. Overall, the project attains certification Level III, which indicates a largely satisfying achievement. It explains that the software processes were implemented quite systematically, however there stil exists some practices with low performance.

**Table-4.** The quality levels for evaluation criteria.

| Software processes | No. of evaluation criteria | Quality Levels |
|---|---|---|
| Agile software process | 31 | III |
| | 4 | IV |
| | 1 | II |
| Secure software process | 20 | III |
| | 16 | IV |

### Feedbacks from the validation

After implementing the ESPAC Model and obtaining the results, the software practitioners were satisfied with the ESPAC Model and agreed that it is practical to be implemented in the real world environment. As mentioned earlier, they validated the ESPAC Model based on three main criteria, which are gain satisfaction, interface satisfaction and task support satisfaction. They are briefly described subsequently.

### Gain satisfaction

The participants pointed out that the proposed model is useful for their working environment. By having this model, they can know the current quality level of their software process. In addition, the outcomes from the assessment and certification process can guide them to improve their software process. Also, the proposed model is deemed as very cost-effective since the assessment is done by the organization internally.

### Interface satisfaction

According to the responses of the participants, the proposed model is perceived as easy to be used because it uses a well-defined processes, activities, and techniques. Additionally, the model is found to be well organized and structured whereby the sequences of the assessment processes and activities are organized in a clear and understandable manner. Moreover, since the assessment involves representative from the assessed team, the assessment can be performed faster and easier since the representative already understood the project well.

### Task support satisfaction

The participants agreed that the proposed model is able to produce usable and expected results. They indicated that they are satisfied with the quality levels and certification level obtained for their projects. They also highlighted that the five factors which are assessed in the proposed model are sufficient to produce relevant results. Besides, from the quality levels obtained, they are able to identify which are the practices that they need to improve. The participants indicated that the proposed model is easy to be implemented, readable and understandable, as it provides a series of activities that can be followed easily. In a nutshell, based on the feedbacks from the software practitioners, the proposed ESPAC Model gained satisfaction of the software practitioners and found as practical to be implemented in the real world projects.

## CONCLUSIONS

The ESPAC Model discussed in this paper has been developed by using the QFD approach. It has addressed the issues faced in the existing software process certification and standards by incorporating agile and secure software processes in its reference standard. Furthermore, the weight values have been included during assessment process which increases the consistency in making decision on the certification results. This has been realized by adapting the AHP technique for weight allocation. The ESPAC Model has been validated by seven software practitioners through a focus group session. The results obtained from the validation demonstrates software practitioners' satisfaction and the practicality of the ESPAC Model to be implemented in the real world environment. Besides assessing the current level of software process being performed by an organization, this model also supports for continuous improvements.

**ACKNOWLEDGEMENTS**

The authors would like to sincerely thank the respondents who participated in this research. Additionally, special thanks go to the Ministry of Higher Education for supporting this research through the FRGS grant (S/O Code: 11889).

**REFERENCES**

Acuna, S. T., Antonio, A. D., Ferre, X., Lopez, M., & Mate, L. (2000). The software process: modeling, evaluation and improvement. In Chang, S. K. Handbook of software engineering and knowledge engineering (pp. 193-237). River Edge: World Scientific Publishing Co. Pte. Ltd.

Agile Manifesto. (2012). Retrieved November 10, 2009, from www.agilemanifesto.org.

Al-Tarawneh, F. H. (2014). A framework for COTS software evaluation and selection for COTS mismatches handling and non-functional requirements. (Unpublished doctoral dissertation). Universiti Utara Malaysia, Malaysia.

Ares, J., Garcia, R., Juristo, N., Lopez, M., Moreno, A.M. (2000). A more rigorous and comprehensive approach to software process assessment. Software Process: Improvement and Practice, 3-30. John Wiley & Sons Ltd.

Aziz Deraman, Jamaiah Yahya, Fauziah Baharom, Amalina Farhi Ahmad Fadzlah & Abdul Razak Hamdan. (2007). Continuous quality improvement in software certification environment. Proceedings of the International Conference on Electrical Engineering and Informatics, 11-17.

Boehm, B., Turner, R. (2005). Management challenges to implement agile processes in traditional development organizations, Software, 30-39. doi: 10.1109/MS.2005.129.

Cater-Steel (2004). An evaluation of software development practice and assessment-based process improvement in small software development firms. (Ph.d Thesis). Retrieved from https://eprints.usq.edu.au/1256/3/Cater-Steel_PhD_%28non_USQ%29_Main_document.pdf
CMMI Product Team (2010). CMMI for Development V1.3 Technical Report.

Davis, N. (2009). Secure software development lifecycle process. Software Engineering Institute. Retreived April 1, 2011 from https://buildsecurityin.uscert.gov/bsi/articles/knowledge/sdlc/326-BSI.html.

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

De Win, B., Scandariato, R., Buyens, K., Gregoire, J., & Joosen, W. (2009). On the secure software development process: CLASP, SDL and Touchpoints compared. Information and software technology, 51(7), 1152-1171.

Evans, R., Tsohou, A., Tryfonas, T., & Morgan, T. (2010). Engineering secure systems with ISO 26702 and 27001. International Conference on System of Systems Engineering, 1-6. doi: 10.1109/SYSOSE.2010.5544065.

Fabbrini, F., Fusani, M., & Lami, G. (2006). Basic concepts of software certification. First International Workshop on Software Certification. Retreived from http://www.cas.mcmaster.ca/sqrl/papers/SQRLreport37.pdf#page=10.

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.

Fauziah Baharom, Aziz Deraman & Abdul Razak Hamdan (2005). A survey on the current practices of software development process in malaysia. Journal of ICT, 57-76.

Hazzan, O., & Dubinsky, Y. (2009). Workshop on human aspects of software engineering. Proceeding of The 24th ACM SIGPLAN Conference Companion on Object Oriented Programming Systems Languages And Applications, 725-726. doi: 10.1145/1639950.1639984.

Heck, P., Klabbers, M., Eekelen, M. (2010). A software product certification model. Software Quality Journal, 18(1) 37-55. doi: 10.1007/s11219-009-9080-0.

Jamaiah Yahya (2007). The development of software certification model based on product quality approach. (Unpublished doctoral dissertation). Universiti Kebangsaan Malaysia, Selangor, Malaysia.

Jones, C., & Bonsignour, O. (2012). The economics of software quality. Boston: Pearson Education.

Kitchenham, B. A., & Pickard, L. M. (1998). Evaluating software engineering methods and tools: part 9: quantitative case study methodology. SIGSOFT Softw. Eng. Notes, 23(1), 24-26. doi: 10.1145/272263.272268.

Kontio, J., Bragge, J., & Lehtola, L. (2008). The focus group method as an empirical tool in software engineering. In Shull, F., Singer, J., & Sjoberg, D. D. K. Guide to advanced empirical software engineering (pp. 93-116). London: Springer-Verlag.

Krueger, R. A. (1994). Focus group a practical guide for applied research. Thousand Oaks: SAGE Publications.

Kunda, D. (2003). STACE: Social technical approach to COTS software evaluation. component-based software quality, In Cechich, A., Piayyini, M., & Vallecillo, A. Component-Based Software Quality (pp. 64-84). Berlin Heidelberg: Springer-Verlag.

Lascelles, D. & Peacock, R. (1996). Self-assessment for business excellence. Berkshire: McGraw-Hill.

Liberatore, M. J., & Nydick, R. L. (1997). Group decision making in higher education using the analytic hierarchy process. In: Liberatore, M. & Nydick, R. L. Research in Higher Education (pp. 593-614). Netherlands: Springer.

Malczewski, J. (1999). GIS and multicriteria decision analysis. New York: John Wiley & Sons.

McConnell, S. (2000). Closing the gap. IEEE. 1 (19).

McGraw, G. (2004). Software security. Security & Privacy, IEEE, 2(2), 80-83. doi: 10.1109/MSECP.2004.1281254.

Merkow, S. M. & Raghavan, L. (2010). Secure and resilient software development. Boca Raton: Auerbach Publications.

Mollaghasemi, M. (1997). Technical briefing: making multiple-objective decisions. California: IEEE Computer Society Press.

O'Regan, G. (2002). A practical approach to software quality. London: Springer.

Pressman, R. S. (2010). Software engineering a practitioner's approach 7th Ed. New York: McGraw-Hill Higher Education.

Rae, A., Robert, P., & Hausen, H.-L. (1995). Software evaluation for certification principles, practice and legal liability. England: McGraw-Hill.

Ritchie, L., & Dale, B. G. (2000). Self-assessment using the business excellence model: a study of practice and process. International Journal of Production Economics, 66(3), 241-254.

Saaty, T. L. (1990). The analytic hierarchy process. New York: McGraw-Hil.

Salo, O. & Abrahamsson, P. (2008). Agile methods in European embedded software development organizations: a survey study of Extreme Programming and Scrum, IET Software, 2(1), 58-64. doi: 10.1049/iet-sen:20070038.

SCAMPI Upgrade Team (2011). Standard CMMI® appraisal method for process improvement (SCAMPISM) A, Version 1.3: Method Definition Document Handbook.

Scriven, M. (1991). Evaluation thesaurus: fourth edition. Sage Publications.

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, 765-774.

Serkani, E. S., Mardi, M., Najafi, E., Jahanian, K., & Herat, A. T. (2013). Using AHP and ANP approaches for selecting improvement projects of Iranian Excellence Model in healthcare sector. African Journal of Business Management, 7(23).

Shafinah Farvin Packeer Mohamed, Fauziah Baharom & Aziz Deraman. (2014a). An exploratory study of agile based software development practices. Journal of Software Engineering and its Application, 8 (5), 85-114.

Shafinah Farvin Packeer Mohamed, Fauziah Baharom & Aziz Deraman. (2014b). Knowledge sharing on implementing software process quality model evaluation: focus group approach. Proceedings of Knowledge Management International Conference.

Sommerville, I. (2007). Software engineering 8th Ed. Harlow: Pearson Education Limited.

Tarí, J. J., & Heras–Saizarbitoria, I. (2012). The self-assessment process and impacts on performance: A case study. International Journal for Quality Research, 6(4). Retreived from http://www.ijqr.net/journal/v6-n4/5.pdf.

Triantaphylluo, E. (2000). Multi-criteria decision making methods: a comparative study. Netherlands: Kluwer Academic Publishers.

Trochim, W. M. (2006). The research methods knowledge base. Retrieved August 6, 2010, from http://www.socialresearchmethods.net/kb/dedind.php.

Vaidya, O.S., & Kumar, S. (2006). Analytic hiearachy process: an overview of applications, European Journal of Operational Research, 169(1), 1-29.

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

Voas, J. (1999). User participation-based software certification. In Vermesan, A. & Coenen, F. Validation and Verification of Knowledge Based Systems, 267-276. US: Springer-Verlag.

Voas, J. (1998). The software quality certification triangle: crosstalk. The Journal of Defense software engineering, 12-14. Retreived from http://www.crosstalkonline.org/storage/issue-archives/1998/199811/199811-Voas.pdf.

Weber-Jahnke, J. H. (2011). A Preliminary Study of Apparent Causes and Outcomes of Reported Failures with Patient Management Software. 3rd Workshop on Software Engineering in Health Care, 5-8.

Yoon, K. & Hwang, C. (1995). Multiple attribute decision-making: an introduction. Sage Publisher.

Zhou, Z., & Liang, K. (2013). Network Course Evaluation System Based on AHP Theory. In Wenjiang, Du. Informatics and Management Science II (pp. 569-575): London: Springer-Verlag.

Zultner, R. E. (1992). Quality function deployment (QFD) for Software. American Programmer.