SOFTWARE PROCESS ASSESSMENT AND CERTIFICATION: APPLICATION OF THE ANALYTIC HIERARCHY PROCESS FOR PRIORITY DETERMINATION

Shafinah Farvin Packeer Mohamed*[1]
School of Computing,
Universiti Utara Malaysia,
Sintok, 06010, Kedah, Malaysia,
shafinah@uum.edu.my

Fauziah Baharom
School of Computing,
Universiti Utara Malaysia,
Sintok, 06010, Kedah, Malaysia,
fauziah@uum.edu.my

Aziz Deraman
Faculty of Ocean Engineering Technology and Informatics,
Universiti Malaysia Terengganu,
Kuala Terengganu, 21300, Terengganu, Malaysia,
a.d@umt.edu.my

Omar Tarawneh
Software Engineering Department,
Amman Arab University,
Amman, Jordan,
o.husain@aau.edu.jo

Yuhanis Yusof
School of Computing,
Universiti Utara Malaysia,
Sintok, 06010, Kedah, Malaysia,
yuhanis@uum.edu.my

ABSTRACT

Software certification involves assessing and certifying the quality of the software process based on multiple evaluation criteria where each criterion has different importance values on the quality of the software. However, the different importance values of the evaluation criteria have not been addressed in the existing software process certification models. A systematic technique is needed to ensure that the certification results are consistent, accurate and not made arbitrarily. To address this issue, the

[1] Corresponding author

ISSN 1936-6744
https://doi.org/10.13033/ijahp.v14i3.870

Extended Software Process Certification (ESPAC) model was introduced by adopting the Analytic Hierarchy Process (AHP) technique to determine the priorities of the evaluation criteria. There were three main phases in this study: (a) theoretical study, (b) expert review and (c) focus group discussion. Ultimately, a mutual agreement was achieved about the evaluation criteria and the AHP was shown to be a suitable technique to be employed in software process assessment and certification. Furthermore, the acquired priorities were used as the ideal priorities for the ESPAC model, which can be used by assessors during the assessment and certification process. The outcome of this study benefits researchers in the AHP and software process assessment fields.

Keywords: software engineering; software certification; ESPAC model; priority; Analytic Hierarchy Process

1. Introduction

The need for software in today's world is constantly on the rise and its usage has become more critical than ever in every domain. Therefore, software developers must be able to produce high quality software in a shorter time to market and rapidly react to changing requirements in order to compete in today's business environment. However, although software developers claim that they produce high quality software, customer dissatisfaction still exists. This is evidenced by continuous reports of software failures that have affected various industries like banking, airlines and even social media platforms (ComputerWorldUKStaff, 2020). These incidents have caused huge loss and disruption of services. Jones and Bonsignour (2012) found that the software failure rate is among the highest compared to other products in recorded human history. Also, a study conducted by The Standish Group discovered 71% of software projects were failures or were being challenged in 2015 (Meier, 2017). Concern about the quality of software has triggered doubt among customers, particularly in terms of investing in such projects.

One way to satisfy customers' requirements on the quality of software is through certification (Ferreira et al., 2019; Pietrantuono & Russo, 2018; Voas & Laplante, 2018; Darwish, 2016; Baharom et al., 2011; Heck et al., 2010). Certification is the process of giving a written assurance that a process, product, or service complies with a criterion and is performed by a third party (Rae et al., 1995). Through software certification, customers can have greater confidence in the quality of the software that they are going to invest in because certification involves independent assessment; therefore, it is assumed that the risk of failure is reduced (Rae et al., 1995; Sun-Jen & Wen-Ming, 2006). A study conducted by Baharom et al. (2005) with software practitioners in Malaysia discovered that software certification is certainly required to confirm the quality of the software. More recently, Ferreira et al. (2019) and Pietrantuono and Russo (2018) highlighted the importance of certification.

According to Voas (1998), the three approaches to certify software's quality are process, product, and personnel. Studies by Heck et al. (2010) and Yahya (2007) focused on the product approach. Similarly, Gualo et al. (2020) worked towards certification of the functional suitability for master data management applications. Nevertheless, the quality of newly developed software cannot be immediately determined by using the product approach because the software needs to be used for a period of time before the quality can be determined (Baharom et al., 2011; Heck et al., 2010). It is different with the process approach, where the software process is assessed from the beginning of the software development. This enables customers to know the capability of an organization to produce high quality software, which will then help them decide whether to invest or not in any of the software development projects. In essence, as mentioned by Voas and Laplante (2018), the best way to certify a software product is through its process. The software process is a "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 et al., 2000, p.1). The underlying idea behind this is that by having a well-defined certified development process, the produced software will be of a guaranteed quality. As highlighted by Deming (1982) and Humphrey (1989), the quality of the end product is determined by the quality of the process employed to produce the product. Therefore, by using the process approach, customers are assured that the software process has been implemented effectively and efficiently.

A considerable amount of literature has been published on models and standards that perform assessments on the quality of the software process, for example, the Capability Maturity Model Integrated (CMMI) (CMMI Institute, 2018) and ISO/IEC 15504 (Mas et al., 2012; Galin, 2004). However, as indicated by Acuna et al. (2000) the aim of these existing models and standards is more to assess and improve the software process rather than provide a mechanism for certification. Therefore, Baharom (2008) developed a certification model that assesses the software process known as Software Process Assessment and Certification (SPAC). However, the existing models are more concerned with the conventional software development process, which emphasizes that each phase of software development must be completed before going on to the next. The requirements in the early stage of software development must be completed before moving onto design, coding and testing. This type of software process also focuses on producing documentation (Sommerville, 2007). However, to survive in today's business environment which demands high quality and secured software, incorporating Agile and secure software processes has become essential in order to produce higher quality software faster (Ansari et al., 2018; Pressman, 2010; Sommerville, 2007).

Therefore, the existing software process and the certification model are enhanced by incorporating the Agile and secure software development processes. The enhanced model is the Extended Software Process Assessment and Certification model (ESPAC) (Packeer Mohamed et al., 2015). Moreover, the existing software process certification models and standards also lack an appropriate synthesis technique. Baharom (2008) observed that the priorities of evaluation criteria are not considered in the existing software process certification models and standards. Priorities of the evaluation criteria need to be considered especially when it has been conclusively shown that an assessment involving multiple criteria will have different importance. Therefore, these criteria should be prioritized (Saaty & De Paola, 2017; Saaty, 2008). Additionally, the process of assigning priorities to the evaluation criteria is significant, particularly when qualitative information is needed from the decision-makers (Triantaphyllou & Mann, 1995). Despite its importance, little attention has been given to the consideration of priorities for the assessed criteria in existing software process assessment and certification. Consequently, this issue was addressed in the ESPAC model in order to produce more consistent and better quality certification results. The software process is assessed based on five main criteria, namely process, people, technology, project constraints, and working environment. These factors are further decomposed into sub-factors and evaluation criteria. The Analytic Hierarchy Process (AHP) was employed as the synthesis technique to obtain the priorities for the evaluation criteria in the ESPAC model (Saaty & De Paola, 2017; Saaty, 2008; Saaty, 1990).

The AHP is one of the most outstanding multi-criteria decision-making (MCDM) techniques. It provides a mechanism for decision-makers to break a problem into a hierarchy, prioritize the evaluation criteria and find a suitable solution for the problem. Unlike other MCDM techniques, the AHP is rooted in determining the priorities of criteria (Saaty & De Paola, 2017; Saaty, 2008; Saaty, 1990). The main benefit of utilizing the AHP technique is that it offers systematic steps to synthesize information through a structured hierarchy. The hierarchy contains the criteria and the sub-criteria, which can help decision-makers understand and simplify a problem by providing better focus during the priority allocation for both the criteria and the sub-criteria (Ishizaka & Labib, 2011). This is the advantage gained in this study as it involves numerous factors, sub-factors, and evaluation criteria. Furthermore, by using the hierarchy, the criteria are systematically organized. In addition, with the use of the AHP, judgment accuracy can be improved because by integrating relative numbers, there should be no or only minimal loss of accuracy (Crostack et al., 2007). More importantly, this technique is appropriate for group decision-making because it allows favorable agreement among the group members (Marjani et al., 2012; Lai et al., 2002). This is important as the study involves group decision-making to determine priorities. Above all, the judgments made in the AHP are more accurate and consistent as this technique provides a mechanism to test the consistency of the judgements. Considering the aforementioned advantages, this study adopted the AHP as its synthesis technique. The seven steps involved in implementing this technique will be explained later. Even though the AHP has been used widely in various areas, this technique has not been applied in the field of software process assessment and certification, and therefore this study contributes to this field of study.

This paper describes the AHP in Section 2 and the methodology of the study in Section 3. A thorough explanation of the priority determination using the AHP is in Section 4. The outcomes of the AHP implementation through the focus group discussion is in Section 5 and ends with the conclusion.

2. Analytic Hierarchy Process

Many techniques can be utilized to make decisions that involve multi-criteria. One of the most utilized techniques is the Analytic Hierarchy Process (AHP) (Saaty & De Paola, 2017; Saaty, 2008; Saaty, 1990). This technique enables decision-makers to represent decision-making problems, which involve multiple criteria in a hierarchy, commonly comprised of three levels. The first level refers to the overall goal for a problem, the second level represents the evaluation criteria, and the third level is comprised of various alternatives. However, in this study, the hierarchy only contains the goal and several levels of evaluation criteria because it does not involve making decisions from among a number of alternatives. The priorities are obtained through pairwise comparisons, which are performed among the evaluation criteria of each level. Then, a normalized ranking is obtained by applying the Eigen value method. Other simpler methods that can be used include the normalization of row average (NRA), normalization of the reciprocal sum of columns (NRC), average of normalized columns (ANC), and normalization of the geometric mean of the rows (NGM) (Hsiao, 2002). This study adopted the NGM method.

The AHP has been applied extensively in various areas such as selection, ranking, and evaluation. For example, Baidya et al. (2018) applied the AHP to select the most appropriate maintenance technique in manufacturing by considering the strategic, planning and operational criteria, while Ali et al. (2018) utilized the AHP to rank suitable sites for wind farm installation using multiple criteria. Kumari and Shylaja (2019) utilized the AHP in the routing protocol. Al-Tarawneh (2014), Kunda (2003), Zhou and Liang (2013), and Chen et al. (2013) evaluated the component-based software, the network course in China, and the potential outsourcing partner, respectively. Padumadasa (2009) utilized the technique to select and evaluate tenders, while Kumar et al. (2019) applied the AHP to assess the quality of soil. More recently, Moradi (2022) applied AHP with Balanced Scorecard and TOPSIS to evaluate the performance of university faculty.

In the field of software process assessment, the AHP has been utilized for software process improvement (Jung, 2001). This technique has also been widely used in combination with other multi-criteria decision-making techniques. For example, the AHP has been applied with PROMETHEE, TOPSIS and WSM (Mokhtar et al., 2017). Animah and Shafiee (2019) applied the AHP with PROMETHEE to select the best strategy for performing maintenance for shipboard machinery systems. Moreover, the AHP was combined with TOPSIS to determine the best fresh fruit bunches in Hambali and Rahman’s (2017) research. More recently, Zaidan et al. (2020) applied the AHP with TOPSIS to select a qualified programmer. These studies used the AHP to obtain the priority and continued with other appropriate multi-criteria decision-making techniques. Similarly, the current study adopted the AHP to obtain the priorities for the evaluation.

3. Methodology

This study was conducted in three main phases, namely, theoretical study, expert review, and a focus group discussion.

3.1 Theoretical study

In this phase, the existing literature including journals, books, proceeding papers and dissertations, were reviewed. The main aim of this phase was to obtain the factors that can influence software quality. The factors were analyzed and classified in order to form a hierarchy that could be used in the AHP implementation. They were classified into five main factors including process, people, project constraints, technology, and environment. Each of these factors was further decomposed into sub-factors and evaluation criteria, as discussed in Section 4 (Step 1).

3.2 Expert review

In this phase, the identified factors were verified through expert review. The main aim of this phase was to verify the factors, sub-factors and the evaluation criteria derived from the theoretical study. The expert review was used because of its simplicity, low cost, and quick completion. Moreover, it is accepted as a significant way to detect and remove defects (Komuro & Komoda, 2008). The following three steps were executed in the expert review:

i. Identify the experts

The experts were chosen from among academicians (knowledge experts) by following the characteristics of experts suggested by Hallowell and Gambatese (2010). The characteristics include: (a) are currently attached to the field of the study under examination, (b) hold an advanced degree (PhD.), (c) are faculty members in an accredited university, (d) authorship, and (e) have at least 5 years of experience. Additionally, software practitioners were also included as experts to perform the verification. Their insight is important since they can give feedback based on their real life experience as domain experts. They were chosen using purposive sampling (Liamputtong, 2011). Four characteristics were used to choose the domain experts: (a) are Agile software practitioners, (b) have experience in secure software process, (d) have more than 3 years of software development experience.

ii. Determine the verification criteria

The factors, sub-factors and evaluation criteria were verified for their comprehensiveness, understandability, accurateness, and organization. These criteria were adapted from previous studies (Al-Tarawneh, 2014; Behkamal et al., 2009; Kunda, 2003). The experts provided their feedback via a checklist.

iii. Collect and analyze the feedback

The experts' feedback was collected and analyzed for further improvement.

3.3 Focus group discussion

The outcomes of the expert review revealed that the factors, sub-factors and evaluation criteria were acceptable. These factors were used to construct the hierarchy for the AHP implementation. To implement the AHP, a focus group discussion was performed by adapting the guidelines from Martakis and Daneva (2013), Mazza and Berre (2007), and Kontio et al. (2008). Focus group discussions have been applied extensively in the software engineering field for evaluation or gleaning practitioners' experiences (Daneva & Ahituv, 2011; Mazza & Berre, 2007; Kontio et al., 2008). The most appropriate participants for this purpose are the software practitioners since they have vast experience with the software process. Three main stages were involved, as depicted in Figure 1.

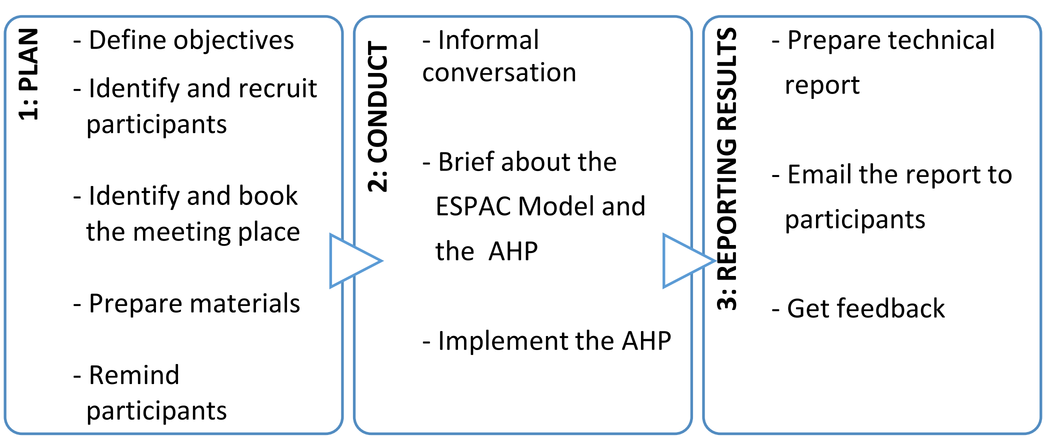


Figure 1 Focus group execution

ISSN 1936-6744
https://doi.org/10.13033/ijahp.v14i3.870

Stage 1. Plan

Proper planning was needed to conduct an effective focus group session. First, the objective of the focus group was determined. The objective was to implement the AHP to determine the priorities of the factors that influence software quality. Then, the participants were identified using purposive sampling based on several common characteristics explained in Section 3.2. The participants were solicited by phone, email and Facebook invitations. Next, the meeting venue was identified, which was a hotel in the Kuala Lumpur area as it was central for all participants. More importantly, the meeting room provided in the hotel was considered neutral since it did not have any influence on the participants, which could affect the discussion. Next, the materials needed for the session were prepared which included presentation slides, documents, cards, certificates of appreciation as well as an Excel file, which automated the priority calculation. Finally, the participants were contacted to remind them about the session. This was to confirm their attendance so that they would not miss the session and let them know that their attendance was important.

Stage 2. Conduct

On the planned date and time, the focus group session was organized and seven participants attended; this is an appropriate number of participants needed to conduct a focus group (Morgan, 1998). Initially, the participants were engaged in an informal conversation to create rapport before the discussion started. This was to make them feel comfortable and relaxed and help the participants get to know each other since they were from different companies. Next, the discussion started with some ice-breaking conversation within the group which further facilitated the moderator and the participants getting to know each other better. This was important to build rapport and create group cohesion (Liamputtong, 2011). A briefing on the objectives of the focus group as well as the ESPAC model, the AHP technique and its implementation in the focus group discussion, then followed. The participants started to communicate and asked questions about the issues that were not clear to them. Then, the AHP was implemented.

This study adopted the group AHP technique, in which decisions on pairwise comparisons were made as a group. To simplify the judging process in a group, the planning poker technique was adapted. This technique is widely used in Agile software development to perform task estimations among software developers (Dyba et al., 2014). The participants were provided with guidelines on the AHP technique and a list of pairwise comparisons. Table 1 shows an excerpt of the pairwise comparisons list. It comprises the factors in level one of the hierarchy. The participants were also provided with nine cards that contained numbers from 1-9 used to make pairwise comparisons.

Table 1
Pairwise comparisons list for factors in level one of the hierarchy

Factors	**Equality**	**Scale**
Process	is [more / equally/ less] important than/to People by a factor of	1 2 3 4 5 6 7 8 9
Process	is [more / equally/ less] important than/to Technology by a factor of	1 2 3 4 5 6 7 8 9
Process	is [more / equally/ less] important than/to Project Constraint by a factor of	1 2 3 4 5 6 7 8 9
Process	is [more / equally/ less] important than/to Environment by a factor of	1 2 3 4 5 6 7 8 9
People	is [more / equally/ less] important than/to Technology by a factor of	1 2 3 4 5 6 7 8 9
People	is [more / equally/ less] important than/to Project Constraint by a factor of	1 2 3 4 5 6 7 8 9
People	is [more / equally/ less] important than/to Environment by a factor of	1 2 3 4 5 6 7 8 9
Technology	is [more / equally/ less] important than/to Project Constraint by a factor of	1 2 3 4 5 6 7 8 9
Technology	is [more / equally/ less] important than/to Environment by a factor of	1 2 3 4 5 6 7 8 9
Project Constraint	is [more / equally/ less] important than/to Environment by a factor of	1 2 3 4 5 6 7 8 9

To implement the planning poker, the moderator raised each of the pairwise comparisons one by one. The participants discussed and exchanged their experiences about the evaluation criteria that were being compared. Then, they used the card to choose the importance value for the pairwise comparison. They kept their chosen value to themselves until everyone had chosen one and after everyone was ready, the cards were revealed concurrently. The value of the pairwise comparison was chosen if a consensus was reached; otherwise, the majority vote was taken. However, a compromise among the group was reached if neither a consensus or majority was reached. If this process was still not successful at choosing a value, then the geometric mean was used to obtain the average. Each of the evaluation criteria were compared and the values were entered into the Excel file, which had been prepared earlier. Therefore, the CR value could be obtained once the pairwise comparisons were made for each pairwise comparison matrix. When there was inconsistency, the judgment process was repeated.

Stage 3. Reporting results

After completing the focus group session, a technical report was prepared and sent to the participants via email. They provided the feedback that they were satisfied with the priorities of the factors that influenced software quality obtained during the focus group discussion. The next section provides an example of the AHP implementation to provide a better understanding of the process and uses an example based on the theoretical study conducted and the AHP implementation during the focus group discussion.

4. Priority determination using the AHP technique

Figure 2 illustrates the seven main steps used to obtain the priorities for the ESPAC model evaluation criteria.

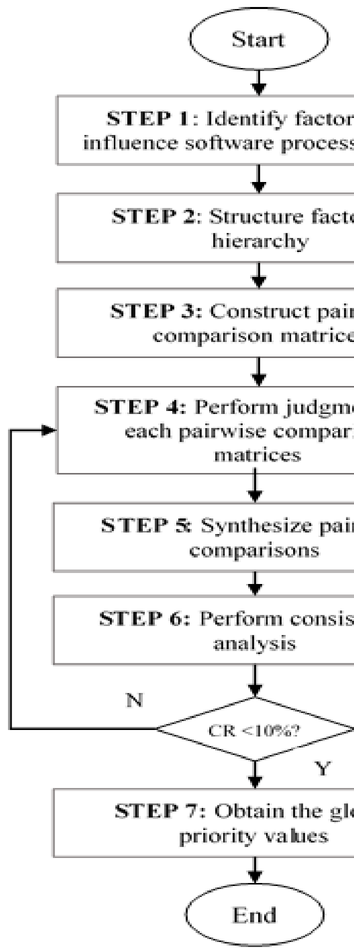


Figure 2 Steps of the AHP to determine priorities

Step 1. Identify the factors that influence software process quality

First, the factors that influence the quality of the software process were determined through the theoretical study. As mentioned earlier, Deming (1982) stated that the quality of the end product is determined by the quality of the process employed to produce the product. Therefore, the software will be good quality as a result of a good quality process. According to Wang and Leung (2001), the software process can be categorized into three perspectives as follows: organization, development and management. Significantly, this highlights that the software process is not simply the methodology followed to develop the software but also the environment where the software is developed. As mentioned by Akbar et al. (2017), the use of the proper software process model greatly influences the software quality that meets customers' needs within the stipulated time and budget. The software process is assessed based on its effectiveness and efficiency. Moreover, according to Gasston (1996), the effectiveness of the software process is based on human,

ISSN 1936-6744
https://doi.org/10.13033/ijahp.v14i3.870

management, economics, and technology. Subsequently, assessing the process alone is not sufficient since the software process is implemented by humans. As argued by Destefanis et al. (2016), developers are one of the key success factors of the software process because they are the ones who are involved throughout the software development. Apparently, with the existence of Agile software development, customers, organizations and developers play an essential role in determining the success of a project in addition to the technology used and the working environment (Aldahmash et al., 2017; Abd El Hameed et al., 2016; Chow & Cao, 2008).

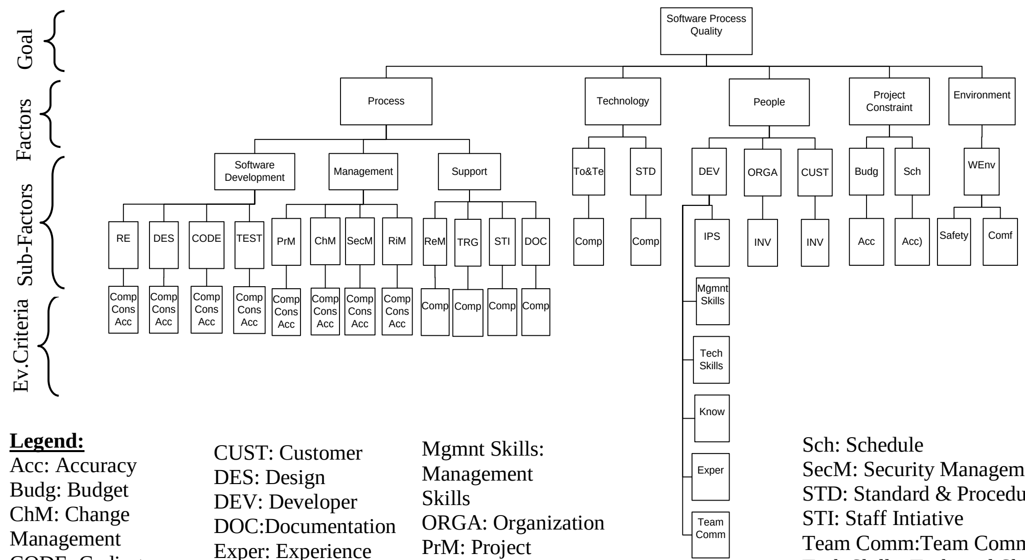
As a result, in this study, other factors than simply the software process that can influence software quality were taken into consideration. The effectiveness of the software process was measured based on the completeness, consistency, and accuracy of the process. The software process must fulfill customers' expectations by having good quality people, appropriate technology, and a stable working environment. On the other hand, efficiency was determined by the capability of producing the software within the expected time and budget (Baharom et al., 2011). Overall, there were five factors used to determine the quality of the software process as listed below:

1. Software process – the quality of the software process carried out when developing the software.
2. People – the quality of the developers, organizations and customers involved with the software development.
3. Technology – the technology used during the software development.
4. Environment - the safety and comfort provided for staff at the location the software is developed.
5. Project constraints – the ability to produce the software within the scheduled time and stipulated budget.

These factors were verified by experts that included three academicians and seven software practitioners. The factors, sub-factors and the evaluation criteria were used for the next step, which was to structure the hierarchy for the AHP implementation.

Step 2. Structure factors in hierarchy

The five factors identified in Step 1 could not be measured directly; therefore, they were decomposed into measurable sub-factors and evaluation criteria as illustrated in Figure 3. They were organized in a hierarchical structure adopted from the AHP technique. The goal was positioned at the first level of the hierarchy followed by the factors, sub-factors and the evaluation criteria in the subsequent levels.



Legend:
Acc: Accuracy
Budg: Budget
ChM: Change Management
CODE: Coding
Comf: Comfort
Comp : Completeness
Cons: Consistency
CUST: Customer
DES: Design
DEV: Developer
DOC:Documentation
Exper: Experience
INV: Involvement
IPS: Interpersonal Skills
Know: Knowledge
Mgmnt Skills: Management Skills
ORGA: Organization
PrM: Project Management
RE: Requirement Engineering
ReM: Resource Management
Sch: Schedule
SecM: Security Management
STD: Standard & Procedure
STI: Staff Intiative
Team Comm:Team Commitment
Tech Skills: Technical Skills
TEST: Testing
To&Te: Tools & Techniques
TRG: Training
WEnv: Working Environment

Figure 3 Hierarchy for software process quality assessment

Step 3. Construct pairwise comparison matrices

To perform the pairwise comparisons, pairwise comparison matrices were constructed for the assessed evaluation criteria. In the pairwise comparison matrix, the sibling criteria at each level were compared in pairs to judge their importance. They were organized in a matrix of two dimensions (square matrix) whereby the compared criteria were sorted vertically in the first column and horizontally in the first row of the matrix.

For example, the five factors that influence the quality of the software located in the first level of the hierarchy (see Figure 3) were compared using one pairwise comparison matrix. The factors were process (proc), technology (tech), people, project constraints (pc), and environment (env). A similar process was performed for the other factors, sub-factors and the evaluation criteria for the entire hierarchy. One pairwise matrix was constructed for each sub-tree with more than one factor/sub-factor/evaluation criteria. There was a total of 18 pairwise comparison matrices as summarized in Table 2.

Table 2
Summary of the pairwise comparison matrices for the ESPAC model

Description	**Level 1**	**Level 2**	**Level 3**	**Level 4**	**Total**
Number of criteria	5	11	26	28	70
Number of pair wise comparison matrices	1	4	5	8	18

ISSN 1936-6744
https://doi.org/10.13033/ijahp.v14i3.870

Step 4. Perform judgments in each pairwise comparison matrix

Judgments must be performed for each two criteria in the matrix. To do this, the relative importance of each two criteria in the matrix was determined using the scale of 1 to 9 created by Saaty (1990). For example, the importance was determined by making the following comparison: C1 is more/equally/less important than/to C2 by a factor of 2/3/4/5/6/7/8/9. The number of pairwise comparisons needed for each matrix was determined by Equation (1).

Pairwise comparisons in each matrix = n (n-1)/2 (1)

where n is the number of criteria in the matrix.

As an example, five factors influence software quality; therefore, n = 5. Accordingly, there were ten pairwise comparisons in this pairwise comparison matrix. This number was obtained using Equation (1), where 5 (5-1) / 2 = 10 pairwise comparisons. Table 3 depicts the pairwise comparison for the first level of the hierarchy, which consists of five criteria. The diagonal elements of the matrix were assigned a value of 1, since $a_{ij}=1$ when i=j. Comparisons were made only for the upper triangular matrix (colored columns), since the lower triangular matrix was comprised of the reciprocals of these values. For example, in the first row of the matrix the process was considered five times more important than the technology and the people, while the process was four times more important than the environment. Additionally, the project constraints were considered equally important as the process. On the other hand, the people and the project constraints were four times more important than the technology, while environment was three times more important than the technology. These are represented in the second row of the matrix.

Table 3
Pairwise comparison factors in first level of hierarchy

Factor	Proc	Tech	People	PC	Env
Proc	1	5	5	1	4
Tech	1/5	1	1/4	1/4	1/3
People	1/5	4	1	1/3	4
PC	1	4	3	1	5
Env	1/4	3	1/4	1/5	1

Step 5. Synthesize pairwise comparisons

Once the pairwise comparisons were complete, they were then synthesized to determine the priority. For this purpose, the Normalization of the Geometric Mean (NGM) was utilized (Hsiao, 2002). The n elements in each row were multiplied, and the n^{th} root was calculated to find the priority. Then, the resulting numbers were normalized. This is shown in Equation (2).

ISSN 1936-6744
https://doi.org/10.13033/ijahp.v14i3.870

$$w_{i}= \left(\prod_{j=1}^{n} a_{ij}\right)^{1/n} / \sum_{i=1}^{n}\left(\prod_{j=1}^{n} a_{ij}\right)^{\frac{1}{n}} \qquad (2)$$

where
w_i = Priority of the evaluation criteria *i*
i = 1,2….,*n*
j = 1,2…..,*n*
a_{ij} = Pairwise comparison in matrix *ij*

Considering the pairwise comparisons made in Table 3 and Equation 2, the priority of the process was calculated as below:

Priority for Process = $(1*5*5*1*4)^{1/5}$ / (2.512+0.334+1.1013+2.268+0.52)
= 2.512 / 6.647
= 0.378

Similarly, the priority was calculated for the remaining factors. Table 4 shows the priorities obtained for the factors in the first level of the hierarchy.

Table 4
Priority for factors in first level of hierarchy

Factor	**Proc**	**Tech**	**People**	**PC**	**Env**	**n^{th} Root values**	**Priority**
Proc	1	5	5	1	4	2.512	0.378
Tech	1/5	1	1/4	1/4	1/3	0.334	0.05
People	1/5	4	1	1/3	4	1.013	0.152
PC	1	4	3	1	5	2.268	0.341
Env	1/4	3	1/4	1/5	1	0.52	0.078
Total						**6.647**	**1.000**

Step 6. Perform consistency analysis

To eliminate inconsistency in the judgments, the Consistency Ratio (CR) was calculated for each of the pairwise comparison matrices. This is the advantage of using AHP, whereby the consistency of the decisions can be revealed. The CR value should be less than 0.1 to indicate consistency (Saaty, 1990). Equations (3) and (4) are used to obtain the CR value:

CR = Consistency Index (CI) / Random Index (RI) (3)

where CI is calculated using this formula:
$CI = (\lambda max – n) / (n-1)$ (4)

where n = number of evaluation criteria in the matrix λmax = the average value of consistency vectors

First, the eigenvalue (λmax) was obtained. The priority of each factor (see Table 4) and the pairwise comparisons were arranged in Table 5. Then, the weighted sum vectors were obtained for each row by multiplying the appropriate priority with the pairwise comparison. The results of the multiplication were then summed to obtain the weighted sum vectors for each row.

Table 5
Calculation of weighted sum vectors

Priority (Proc)	**PWC (Proc)**	**Priority (Tech)**	**PWC (Tech)**	**Priority (People)**	**PWC (People)**	**Priority (PC)**	**PWC (PC)**	**Priority (Env)**	**PWC (Env)**	**Weighted sum vectors**
	1		5		5		1		4	2.041
	1/5		1		1/4		1/4		1/3	0.275
0.378	1/5	0.05	4	0.152	1	0.34 1	1/3	0.078	4	0.853
	1		4		3		1		5	1.765
	1/4		3		1/4		1/5		1	0.429

Next, the consistency vectors were calculated by dividing the weighted sum vectors with the respective priority and are shown in Table 6.

Table 6
Calculation of consistency vectors

Criteria	**Weighted sum vectors**	**Priority**	**Consistency vectors**
Proc	2.041	0.378	5.399
Tech	0.275	0.05	5.5
People	0.853	0.152	5.612
PC	1.765	0.341	5.176
Env	0.429	0.078	5.5

Then, the average of the consistency vectors was calculated to obtain $\lambda_{max.}$ Next, the CI value was calculated. Equation 4 was used to obtain the CI value. Finally, Equation 3 was utilized to obtain the CR value. The RI value was determined based on the value of n, as provided by Saaty (1990). For the current pairwise comparison matrix, since the n value was 5, and RI was 1.12, the CR value obtained is 0.098. Since the CR value was less than 0.1, the pairwise comparison made was considered to be consistent. However, if the CR value was greater than 0.1, then the judgments on the criteria would be considered inconsistent and need to be performed again (return to Step 4). The CR value for each pairwise comparison matrix must be consistent before the global priority can be obtained.

Step 7. Obtain the global priority

The priorities attained from the preceding steps are the local priorities. The final priorities are known as the global priority, and are acquired by multiplying the local priority of a child by its parents' local priority (the calculation starts from the lowest level and goes to the first level of the hierarchy). Equation (5) shows how to obtain the global priority.

$$GW_i = LW_i * \prod_{j=1}^{n} P_j \quad (5)$$

where:

GWi = Global priority for i^{th} evaluation criteria
LWi = Local priority for i^{th} evaluation criteria
Pj = Local priority for j^{th} parents
i = 1,2……,n
j = 1,2……,n

For example, Equation 5 is used to obtain the global priority for the completeness of requirement engineering (GWCompRE). The local priority for the completeness of requirement engineering is multiplied with the local priorities of its parents (requirement engineering, software development, and process) as illustrated in Figure 4.

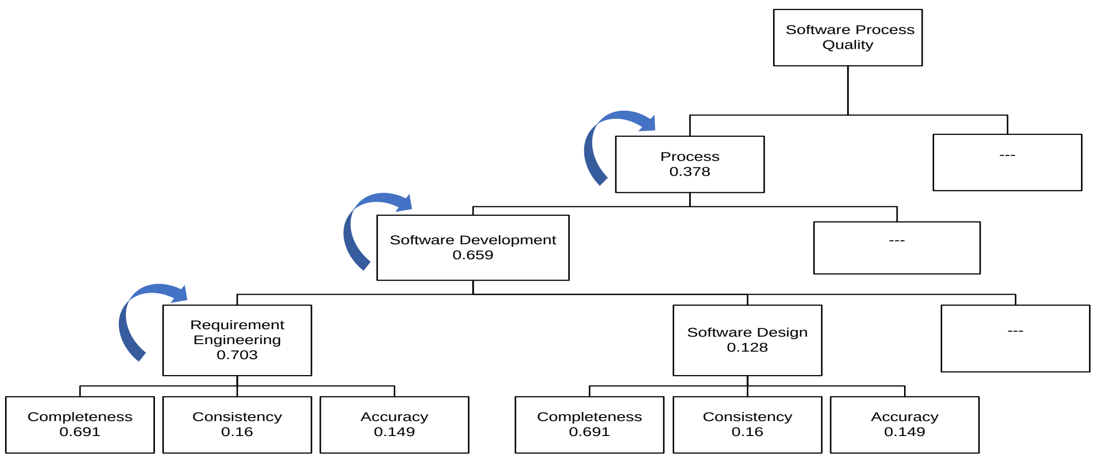


Figure 4 Part of hierarchy with local priority

Likewise, this calculation is performed to obtain the global priority for other factors/sub-factors/evaluation criteria. The complete local priorities and the global priorities are listed in Table 8 (Section 5). These global priorities are the ideal priorities suggested by the ESPAC model. The global priority for completeness of requirement engineering is calculated as:

GWCompRE = 0.691 *0.703 * 0.659 * 0.378
= 0.121

5. Outcomes from the AHP implementation

This section discusses the outcomes obtained from the AHP implementation through the focus group, namely the participants' background and the priorities obtained.

5.1 Participants' backgrounds

The seven participants were Agile practitioners from distinct organizations located in the Kuala Lumpur area. They had different positions, such as team leader and Scrum Master. The participants had more than three years of experience in software development. Four of them worked in large companies with more than 250 employees. Most of them seemed to favor Scrum and Extreme Programming (XP). Table 7 recapitulates the background of the participants.

Table 7
Participants' background

ID	Positions	Size of organization	Experience in software development	Agile methods used
A	Team leader	>250	11-20	FDD
B	Team leader	51-250	6–10	Scrum
C	Architect	>250	11-20	Scrum, XP
D	Programmer	51-250	6–10	Scrum, XP
E	Programmer	>250	6-10	Scrum, XP
F	Application lifecycle manager	20-50	11-20	Scrum, TDD
G	Scrum master	>250	6-10	Scrum, XP, AM, Lean

5.2 Priorities obtained

The priorities that were obtained through the planning poker technique are provided in Table 8. The table consists of the local priorities for all the factors/sub-factors/evaluation criteria as well as the global priorities. The global priorities are used as the ideal priority for the ESPAC model, which can be used by potential users of the model. However, if the assessors do not agree with those values, they can perform the AHP technique again and obtain their own priorities, which might be more suitable for their organizations.

Table 8
Priority values

Factors / Sub-factors / (Local priorities)			Evaluation criteria	Global / Ideal priorities
Process	Software development (0.659)	Requirement engineering (0.703)	Completeness (0.691)	0.121
			Consistency (0.16)	0.028
			Accuracy (0.149)	0.026
		Software design (0.128)	Completeness (0.691)	0.022
			Consistency (0.16)	0.005
			Accuracy (0.149)	0.005
		Coding (0.071)	Completeness (0.691)	0.012
			Consistency (0.16)	0.003
			Accuracy (0.149)	0.003
		Testing	Completeness (0.691)	0.017

ISSN 1936-6744
https://doi.org/10.13033/ijahp.v14i3.870

Factors / Sub-factors / (Local priorities)			**Evaluation criteria**	**Global / Ideal priorities**
(0.378)		(0.097)	Consistency (0.16)	0.004
			Accuracy (0.149)	0.004
	Manage-ment (0.156)	Project management (0.25)	Completeness (0.333)	0.005
			Consistency (0.333)	0.005
			Accuracy (0.333)	0.005
		Change management (0.25)	Completeness (0.333)	0.005
			Consistency (0.333)	0.005
			Accuracy (0.333)	0.005
		Security management (0.25)	Completeness (0.333)	0.005
			Consistency (0.333)	0.005
			Accuracy (0.333)	0.005
		Risk management (0.25)	Completeness (0.333)	0.005
			Consistency (0.333)	0.005
			Accuracy (0.333)	0.005
	Support 0.185	Staff initiative (0.499)	Completeness (1.0)	0.035
		Documentation (0.067)	Completeness (1.0)	0.005
		Resource management (0.249)	Completeness (1.0)	0.017
		Training (0.185)	Completeness (1.0)	0.013
Technology 0.05		Tools & techniques (0.2)	Completeness (1.0)	0.01
		Standard & procedures (0.8)	Completeness (1.0)	0.04
People 0.152		Developers (0.084)	Interpersonal skills (0.039)	0.00049
			Management skills (0.054)	0.00069
			Technical skills (0.169)	0.00216
			Knowledge (0.166)	0.00212
			Experience (0.164)	0.00209
			Team commitment (0.409)	0.00522
		Organization (0.211)	Involvement (1.0)	0.03207
		Customers (0.705)	Involvement (1.0)	0.10716
Project constraint 0.341		Budget (0.5)	Accuracy (1.0)	0.1705
		Schedule (0.5)	Accuracy (1.0)	0.1705

Factors / Sub-factors / (Local priorities)		**Evaluation criteria**	**Global / Ideal priorities**
Environment 0.078	Working environment (1.0)	Safety (0.8)	0.062
		Comfort (0.2)	0.016

Based on the opinions of the participants, the evaluation criteria utilized in this study were sufficient to assess the software process. According to Table 8, the most important factor in producing high quality software is the development process, followed by the project constraints, technology, environment, and people in descending order. Even though in the Agile software process people are the main contributors for producing high quality software (Aldahmash et al., 2017; Abd El Hameed et al., 2016; Chow & Cao, 2008), the participants of this study mutually agreed that the other factors were more important. Furthermore, software development was rated as the highest priority and was followed by the support and the management processes. This is in line with the work of Akbar et al. (2017), where the software process was considered the most important to ensure software quality. For the software development phases, the requirement engineering phase had the highest priority and was followed by the software design and testing. Coding had the lowest priority. Requirement engineering was considered highly important because the subsequent steps in software development depend on the correctness of the requirements (Pressman, 2010). The requirements are considered as the groundwork in software production, which will be used to verify the users' requirements and act as an indicator to show that the proposed requirements are fulfilled (Heikkil¨a et al., 2015).

The management processes, which are project management, risk management, security management, and change management, were all rated with equal priority. The management of a project as well as the software development is essential (Yaghoobi, 2018; Ahimbisibwe et al., 2017) not only in reference to managing the project but also the risk, security and change management. Therefore, they are equally prioritized. For the development process, which consists of software development, management and support processes, the completeness criterion obtained the highest priority. This shows that the participants emphasized completing the process while developing the software. Moreover, for the support process, the participants gave the highest importance to staff initiatives, followed by resource management, training, and documentation. The least priority was given to documentation. This might be because they were more concerned about creating the end product. One value of the Agile Manifesto (2001) is "working software over comprehensive documentation". However, even though Agile software development focuses on producing working software, minimal documentation is still needed for further references and maintenance. Wagenaar et al. (2018) concluded that documents in the Agile software development are essential to providing governance to the team. A means of internal communication could be used for quality control and might be needed by other parties as a reference.

Additionally, for people who were involved during the software development, the customers were the highest priority, followed by the organization, and the lowest was software practitioners. This shows that the participants' focus on customer satisfaction aligns with the Agile principles (Agile Manifesto, 2001). Similarly, the outcome from a study conducted by Sambinelli and Borges (2019) revealed that when customers were valued during software development, they observed an improvement in the customers' communication, satisfaction, and collaboration. This can help increase customer's satisfaction and loyalty to the organization. For software practitioners, team commitment was rated as the most important, followed by technical skills, knowledge, experience, management skills, and interpersonal skills. This is because the commitment given by each team member is essential to the production of high quality software since the team produces the software (Poth et al., 2020; Sjøberg, 2018). On the other hand, technical skills, knowledge, experience, and management skills can be gained over time and can be improved through life-long learning processes.

For technology, the completeness of standards and procedures was rated less important than the completeness of tools and techniques. The importance of using tools was described in Ciancarini et al. (2019) where the developers were reported to use tools widely to support software development. However, as mentioned by Salo and Abrahamsson (2008), following proper standards and procedures is essential; otherwise, understanding the requirements might be more difficult. Chow and Cao (2008) also identified that having coding standards upfront as one of the success factors in the Agile software development. Nevertheless, when proper tools are collectively used in a team, the standards and procedures might be indirectly implemented in some way. Furthermore, accuracy of the budget and the schedule were rated as equally important. Both the budget and the schedule are dependent on each other in the production of high quality software and always become a measure of a successful project as widely indicated by researchers (Nath et al., 2020; Alaidaros, & Omar, 2017; Lee & Xia, 2010). On the other hand, safety was considered more important than comfort in the working environment. The working environment is considered one of the motivational tools to increase software developers' productivity (Machuca-Villegas et al., 2020).

Overall, the majority of the priorities or levels of importance ranked by the participants of the focus group have been supported by previous studies. Priorities can differ from one organization to another since they are influenced by the organization's standard of operation and the experience of the developers. Therefore, in the case of implementing the ESPAC model, it is appropriate to let the assessors decide whether to utilize the proposed ideal priorities or to generate new ones by implementing the AHP technique. The priority values are included in the software process assessment for each of the evaluation criteria, which finally produces the certification level and the quality levels (Packeer Mohamed et al., 2015).

5.3 Sensitivity analysis

The sensitivity analysis was performed by using a 'What-if' analysis in order to analyze the changes that might happen to global priority values if the local priority values of the evaluation criteria are changed. With the aid of a sensitivity analysis, the reliability of the initial decision and factors that influenced the result can be determined such as which criteria influenced the original results (Mu & Pereyra-Rojas, 2017). Table 8 shows that the highest local priority value was the process (0.378) factor when compared to the other four factors. Within the sub-factors of process, the highest local priority was the software development process (0.0659) while requirement engineering had the highest priority (0.703) compared to the other software development processes. In order to determine if changes in the local priority values for the requirement engineering has an effect on its global priority, a sensitivity analysis on the completeness, consistency and accuracy criteria was performed.

ISSN 1936-6744
https://doi.org/10.13033/ijahp.v14i3.870

Tables 9 to 11 show the changes in the global priority values. Note that in Table 9, the obtained global priority value for the completeness of requirement engineering is 0.121 while the local priority value is 0.691. In this sensitivity analysis, the local priority value was changed by increasing and decreasing it by 0.005. The result shows that there was only a slight difference (by 0.001) in the global priority value when the local priority value was changed.

Table 9
Changes in global priority values when the local priority for completeness of requirement engineering was changed

Local prioriy	0.121
0.676	0.118
0.681	0.119
0.686	0.120
0.691	0.121
0.696	0.122
0.701	0.123
0.706	0.124

Table 10 depicts the global and local priority values for the consistency of requirement engineering where the initial obtained values are 0.001 and 0.028, respectively. By increasing and reducing the local priority value by 0.005, a small difference in the global priority value was noticed.

Table 10
Changes in global priority values when the local priority for consistency of requirement engineering was changed

Local priority	0.028
0.001	0.000
0.006	0.001
0.011	0.002
0.016	0.003
0.021	0.004
0.026	0.005
0.031	0.005

Table 11 shows the difference of the global priority of the accuracy of requirement engineering when changes were made to the initial local priority. Similar to the earlier outcome, only a small difference was found in the global priority, which is 0.001.

ISSN 1936-6744
https://doi.org/10.13033/ijahp.v14i3.870

Table 11
Changes in the global priority values when the local priority for accuracy of requirement engineering was changed

Local priority	0.026
0.134	0.023
0.139	0.024
0.144	0.025
0.149	0.026
0.154	0.027
0.159	0.028
0.164	0.029

6. Conclusion

This paper presented the technique used to determine the priorities for the evaluation criteria in assessing and certifying the software process. The existing software process assessment and certification models do not consider priorities of the evaluation criteria even though multiple criteria are involved in the assessment. To produce more accurate and consistent certification results, priorities of the evaluation criteria need to be determined in a systematic way. Therefore, this study adopted the Analytic Hierarchy Process (AHP) in the Extended Software Process Certification model (ESPAC) to determine the priorities. Five main factors were used to assess the software process quality, namely process, technology, people, project constraints and environment. These factors were obtained through theoretical study and verified through expert reviews. To obtain the priorities for these criteria, seven steps of the AHP were performed in a focus group discussion with seven software practitioners. The results from the expert reviews and focus group discussion revealed that the experts were satisfied with the evaluation criteria and the AHP technique implemented in this study. Additionally, the priorities obtained were used as the ideal priorities in the ESPAC model. Among the five factors prioritized, process was rated the most important criterion, followed by the project constraints, technology, environment, and people. With the obtained priorities, potential users of this model may utilize the proposed values, or they can obtain preferred priorities using the AHP if the ideal priority is not suitable to their organization’s environment. At the end of this study, a sensitivity analysis was carried out using a 'What-if' analysis to examine the effect on the global priority values if the local priority values of the evaluation criteria were changed. To perform this analysis, the local priority values for completeness, consistency and accuracy of requirement engineering were changed and observed. This analysis revealed that there was only a small difference in the global priority values when the local priority values were changed. This was predicted, and indicates that the initial priorities are reliable. This study contributes to the body of knowledge on the application of the AHP technique in the field of software process assessment and certification. Furthermore, a systematic technique to determine the priorities for the evaluation criteria is provided for the assessors. As for future work, this research can be enhanced by using AI techniques such as fuzzy or firefly algorithms to reduce dependency on human judgment while simultaneously being more transparent.

REFERENCES

Abd El Hameed, T., Latif, M. A. E., & Kholief, S. (2016). Identify and classify critical success factors of agile software development methodology using mind map. *International Journal of Advanced Computer Science and Applications*, *7*(5), 83-92. Doi: https://doi.org/10.14569/ijacsa.2016.070513

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. Doi: https://doi.org/10.1142/9789812389718_0011

Agile Manifesto. (2001). Retrieved from www.Agilemanifesto.org.

Ahimbisibwe, A., Daellenbach, U., & Cavana, R. Y. (2017). Empirical comparison of traditional plan-based and agile methodologies. *Journal of Enterprise Information Management*, *30*(3), 400-454. Doi: 10.1108/JEIM-06-2015-0056

Akbar, M. A., Sang, J., Khan, A. A., Shafiq, M., Hussain, S., Hu, H., & Xiang, H. (2017). Improving the quality of software development process by introducing a new methodology–AZ-model. *IEEE Access*, *6*, 4811-4823. Doi: 10.1109/ACCESS.2017.2787981

Alaidaros, H., & Omar, M. (2017). Software project management approaches for monitoring work-in-progress: A review. *Journal of Engineering and Applied Sciences*, *12*(15), 3851-3857.

Aldahmash, A., Gravell, A. M., & Howard, Y. (2017). A review on the critical success factors of agile software development. *European Conference on Software Process Improvement*, 504-512. Springer, Cham. Doi: 10.1007/978-3-319-64218-5_41

Ali, Y., Butt, M., Sabir, M., Mumtaz, U., & Salman, A. (2018). Selection of suitable site in Pakistan for wind power plant installation using analytic hierarchy process (AHP). *Journal of Control and Decision*, *5*(2), 117-128. Doi: 10.1080/23307706.2017.1346490

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, Kedah, Malaysia.

Animah, I., & Shafiee, M. (2019). Maintenance strategy selection for critical shipboard machinery systems using a hybrid AHP-PROMETHEE and cost benefit analysis: A case study. *Journal of Marine Engineering & Technology*, 1-12. Doi: 10.1080/20464177.2019.1572705

Ansari, M. T. J., Pandey, D., & Alenezi, M. (2018). Store: Security threat oriented requirements engineering methodology. *Journal of King Saud University-Computer and Information Sciences, 34*(2), 191-203. Doi: 10.1016/j.jksuci.2018.12.005

Baharom, F., Yahya, J., Deraman, A., & Hamdan, A. R. (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

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

Baharom, F., Deraman, A., & Hamdan, A. R. (2005). A survey on the current practices of software development process in Malaysia. *Journal of ICT*, *4*, 57-76.

Baidya, R., Dey, P. K., Ghosh, S. K., & Petridis, K. (2018). Strategic maintenance technique selection using combined quality function deployment, the analytic hierarchy process and the benefit of doubt approach. *The International Journal of Advanced Manufacturing Technology*, *94*(1-4), 31-44. Doi: 10.1007/s00170-016-9540-1

Behkamal, B., Kahani, M., & Akbari, M. K. (2009). Customizing ISO 9126 quality model for evaluation of B2B applications. *Information and Software Technology*, *51*(3), 599-609. Doi: 10.1016/j.infsof.2008.08.001

Chen, J. K., Pham, V. K., & Yuan, B. J. (2013). Adopting AHP approach on evaluation and selection of outsourcing destination in East and Southeast Asia. *Technology Management in the IT-Driven Services (PICMET),* 528-537. Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6641715

Chow, T., & Cao, D. B. (2008). A survey study of critical success factors in agile software projects. *Journal of Systems and Software*, *81*(6), 961-971. Doi: 10.1016/j.jss.2007.08.020Get

Ciancarini, P., Missiroli, M., & Sillitti, A. (2019). Preferred tools for agile development: A sociocultural perspective. *International Conference on Objects, Components, Models and Patterns,* 43-58. Springer, Cham. Doi: 10.1007/978-3-030-29852-4_3

CMMI Institute. (2018). *CMMI Adoption and Transition Guidance V2.0*. Retrieved from https://cmmiinstitute.com/getattachment/5868888b-5f37-4715-bc8b-c43250ec0abc/attachment.aspx

ComputerWorldUKStaff. (2020, Feb 17). *Top software failures in recent history*. ComputerWorld. Retrieved from https://www.computerworld.com/article/3412197/top-software-failures-in-recent-history.html#slide1

Crostack, H. A., Hackenbroich, I., Refflinghaus, R., & Winter, D. (2007). Investigations into more exact weightings of customer demands in QFD. *Asian Journal on Quality*, *8*(3), 71-80. Doi: http://dx.doi:org/10.1108/15982688200700026

Daneva, M., & Ahituv, N. (2011). What practitioners think of inter-organizational ERP requirements engineering practices: Focus group results. *International Journal of Information System Modeling and Design*, *2*(3), 49-74. Doi: 10.4018/jismd.2011070103

Darwish S. M. (2016). Software test quality rating: A paradigm shift in swarm computing for software certification. *Knowledge-Based Systems*, 10, 167-175. Doi: 10.1016/j.knosys.2016.07.022

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

Destefanis, G., Ortu, M., Counsell, S., Swift, S., Marchesi, M., & Tonelli, R. (2016).

Software development: Do good manners matter?. *PeerJ Computer Science*, *2*, e73. Doi: 10.7717/peerj-cs.73.

Dyba, T., Dingsoyr, T., & Moe, N. B. (2014). Agile project management. In Ruhe, G. & Wohlin, C. (Eds.), *Software Project Management in a Changing World* (pp. 277-300). Springer Berlin Heidelberg.

Ferreira, G., Kästner, C., Sunshine, J., Apel, S., & Scherlis, W. (2019). Design Dimensions for Software Certification: A Grounded Analysis. arXiv preprint arXiv:1905.09760.

Galin, D. (2004). *Software quality assurance*. England: Pearson Education Limited.

Gasston, J. L. (1996). Process improvement: An alternative to BPR for software development organizations. *Software Quality Journal, 5*, 171-183. Doi: https://doi.org/10.1007/bf00678582

Gualo, F., Caballero, I., & Rodriguez, M. (2020). Towards a software quality certification of master data-based applications. *Software Quality Journal*, *28*, 1-24. Doi: 10.1007/s11219-019-09495-w

Hallowell, M. R., & Gambatese, J. A. (2010). Qualitative research: Application of the Delphi method to CEM research. *Journal of Construction Engineering and Management, 136*(1), 99-107. Doi: 10.1061/_ASCE_CO.1943-7862.0000137

Hambali, A., & Rahman, M. A. (2017). Application of integrated AHP and TOPSIS techniques for determining the best Fresh Fruit Bunches (FFB). *Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 9*(3), 145-149.

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

Heikkilä, V. T., Damian, D., Lassenius, C., & Paasivaara, M. (2015). *A mapping study on requirements engineering in agile software development*. 41st Euromicro Conference on Software Engineering and Advanced Applications, 199-207. IEEE.

Hsiao, S. W. (2002). Concurrent design method for developing a new product. *International Journal of Industrial Ergonomics, 29*(1), 41-55. Doi: 10.1016/S0169-8141(01)00048-8

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

Ishizaka, A., & Labib, A. (2011). Review of the main developments in the Analytic Hierarchy Process. *Expert Systems with Applications, 38*(11), 14336-14345. Doi: 10.1016/j.eswa.2011.04.143

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

Jung, H. W. (2001). Rating the process attribute utilizing AHP in SPICE-based process assessments. *Software Process: Improvement and Practice, 6*(2), 111-122. Doi: 10.1002/spip.139

Komuro, M., & Komoda, N. (2008). *An explanation model for quality improvement effect of peer reviews.* International Conference on Computational Intelligence for Modelling Control & Automation, 1159-1164. Doi: 10.1109/CIMCA.2008.187

Kontio, J., Bragge, J., & Lehtola, L. (2008). The focus group method as an empirical tool in software engineering. In Shull, F. et al. (Eds.) *Guide to advanced empirical software engineering* (pp. 93-116). Springer, London.

Kumar, U., Kumar, N., Mishra, V. N., & Jena, R. K. (2019). Soil quality assessment using Analytic Hierarchy Process (AHP): A case study. In Sugumaran, V. (Ed.), *Interdisciplinary approaches to information systems and software engineering* (pp. 1-18). IGI Global. Doi: 10.4018/978-1-5225-7784-3.ch001

Kumari, N. D., & Shylaja, B. S. (2019). AMGRP: AHP-based multimetric geographical routing protocol for urban environment of VANETs. *Journal of King Saud University-Computer and Information Sciences*, *31*(1), 72-81. Doi: 10.1016/j.jksuci.2017.01.001

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

Lai, V. S., Wong, B. K., & Cheung, W. (2002). Group decision making in a multiple criteria environment: A case using AHP in software selection. *European Journal of Operational Research*, 137, 134-144. Doi: 10.1016/S0377-2217(01)00084-4

Lee, G., & Xia, W. (2010). Toward agile: An integrated analysis of quantitative and qualitative field data on software development agility. *MIS Quarterly*, *34*(1), 87-114. Doi: 10.2307/20721416

Liamputtong, P. (2011). *Focus group methodology principles and practices*. London: Sage Publication.

Machuca-Villegas, L., Gasca-Hurtado, G. P., Tamayo, L. M. R., & Puente, S. M. (2020). Social and human factor classification of influence in productivity in software development teams. *European Conference on Software Process Improvement*, 717-729. Springer, Cham. Doi: 10.1007/978-3-030-56441-4_54

Marjani, M. E., Soh, K. G., Majid, M., Mohd Sofian, O. F., Nur Surayyah, M. A., & Mohd Rizam, A. B. (2012). Usage of group decision making approach in karate agility test selection. *International Symposium on the Analytic Hierarchy Process*, 1-11. Retrieved from http://www.isahp.org/uploads/59.pdf

Martakis, A., & Daneva, M. (2013). Handling requirements dependencies in Agile projects: A focus group with Agile software development practitioners. *Seventh International Conference on Research Challenges in Information Science*, 1-11. Doi: 10.1109/RCIS.2013.6577679

Mas, A., Fluxa, B., & Amengual, E. (2012). Lessons learned from an ISO/IEC 15504 SPI programme in a company. *Journal of Software: Evolution and Process*, *24*(5), 493-500. Doi: 10.1002/smr.501

Mazza, R., & Berre, A. (2007). *Focus group methodology for evaluating information* *visualization techniques and tools*. 11th International Conference Information Visualization, 74-80. Doi: 10.1109/IV.2007.51

Meier, S. R. (2017). *Technology Portfolio Management for Project Managers*. Retrieved from https://www.pmiwdc.org/sites/default/files/presentations/201703/PMIW_LocalCommunity_Tysons_presentation_2017-02.pdf

Mu, E., & Pereyra-Rojas, M. (2017). Understanding the analytic hierarchy process. In Mu and Pereya-Rojas (Eds.), *Practical decision making* (pp. 7-22). Springer, Cham.

Mokhtar, M. R., Abdullah, M. P., Hassan, M. Y., & Hussin, F. (2017). Comparative study of multiple criteria decision making methods for selecting the best demand side management options. *ELEKTRIKA-Journal of Electrical Engineering, 16*(1), 11-16. Doi: 10.11113/elektrika.v16n1.15

Moradi, N. (2022). Performance evaluation of university faculty by combining BSC, AHP, and TOPSIS: from the students'perspective. *International Journal of the Analytic Hierarchy Process, 14*(2), 1-29. Doi: https://ijahp.org/index.php/IJAHP/article/view/915

Morgan, D. L. (1998). *Planning focus groups.* Thousand Oaks: Sage Publications.

Nath, U. K., Jagadev, A. K., & Pattnaik, P. K. (2020). Achieving target schedule time faster through Agile simulation model. *Journal of Critical Reviews*, *7*(19), 4080-4088. Doi: 10.31838/jcr.07.19.476

Packeer Mohamed, S. F., Baharom, F., & Deraman, A. (2015). ESPAC model: Extended Software Process Assessment and Certification model for Agile and secure software processes. *ARPN Journal of Engineering and Applied Sciences, 10*(3), 1364-1373.

Padumadasa, E. U., Colombo, S., & Rehan, S. (2009). Investigation into decision support systems and multiple criteria decision making to develop a web-based tender management system. *International Symposium on the Analytic Hierarchy Process*, 1-17. Retrieved from http://www.isahp.org/2009Proceedings/Final_Papers/66_Padumadasa_EvaluatingTenderOffers_REV_FIN.pdf

Pietrantuono, R., & Russo, S. (2018). *Robotics software engineering and certification: Issues and challenges*. IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), 308-312. IEEE. Doi: 10.1109/ISSREW.2018.00023

Poth, A., Kottke, M., & Riel, A. (2020). Evaluation of agile team work quality. *International Conference on Agile Software Development*, 101-110. Springer, Cham. Doi: 10.1007/978-3-030-58858-8_11

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.

Saaty, T. L., & De Paola, P. (2017). Rethinking design and urban planning for the cities of the future. *Buildings*, *7*(3), 76. Doi: 10.3390/buildings7030076

Saaty, T. L. (2008). Decision making with the Analytic Hierarchy Process. *International Journal of Services Sciences*, *1*(1), 83-98. Doi: 10.1504/IJSSci.2008.01759

Saaty, T. L. (1990). How to make a decision: The Analytic Hierarchy Process, *European Journal of Operation Research, 48*(1), 9–26. Doi: 10.1016/0377-2217(90)90057-I

Sambinelli, F., & Borges, M. A. F. (2019). *Survey on strategies to increase customer value in Brazilian Agile software development companies*. 14th Iberian Conference on Information Systems and Technologies (CISTI*)*, 1-7. IEEE. Doi: 10.23919/CISTI.2019.8760653

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

Sjøberg, D. I. (2018). Teamwork quality and team performance: Exploring differences between small and large Agile projects. *Agile Processes in Software Engineering and Extreme Programming*, 267.

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

Sun-Jen, H., & Wen-Ming, H. (2006). Selection priority of process areas based on CMMI continuous representation. *Information & Management*, *43*(3), 297-307. Doi: 10.1016/j.im.2005.08.003

Triantaphyllou, E., & Mann, S. H. (1995). Using the Analytic Hierarchy Process for decision making in engineering applications: Some challenges. *International Journal of Industrial Engineering: Applications and Practice*, *2*(1), 35-44. Retrieved from http://bit.csc.lsu.edu/trianta/Journal_PAPERS1/AHPapls1.pdf

Voas, J., & Laplante, P. A. (2018). IoT's certification quagmire. *Computer*, *51*(4), 86-89. Doi: 10.1109/MC.2018.2141036

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

Wagenaar, G., Overbeek, S., Lucassen, G., Brinkkemper, S., & Schneider, K. (2018). Working software over comprehensive documentation–Rationales of Agile teams for artefacts usage. *Journal of Software Engineering Research and Development*, *6*(1), 7. Doi: 10.1186/s40411-018-0051-7

Wang, Y., & Leung, H. K. (2001). A benchmark-based adaptable software process model. *Proceedings 27th EUROMICRO Conference. 2001: A Net Odyssey*, 216-224.

Yaghoobi, T. (2018). Prioritizing key success factors of software projects using fuzzy AHP. *Journal of Software: Evolution and Process*, *30*(1), e1891. Doi: 10.1002/smr.1891

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

Zaidan, A. A., Zaidan, B. B., Alsalem, M. A., Momani, F., & Zughoul, O. (2020). Novel multiperspective hiring framework for the selection of software programmer applicants based on AHP and group TOPSIS techniques. *International Journal of Information Technology & Decision Making*, *19*(3), 775-847. Doi: 10.1142/S0219622020500121

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 London. Doi: 10.1007/978-1-4471-4811-1_73