Journal of Advanced Zoology
ISSN: 0253-7214
Volume 45 Issue S1 Year 2024 Page 65:78

# Adaptive Security Activities Selection Model Using Multi-Criteria Decision-Making Methods

**Mazni Mohamed Jakeri[1]*, Mohd Fadzil Hassan[2], Aliza Sarlan[3], Amirudin Abdul Wahab[4]**

*[1]*Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, 32610 Seri Iskandar, Perak Darul Ridzuan, Malaysia. Email: maznym@yahoo.com*
*[2]Institute of Autonomous Systems, Universiti Teknologi PETRONAS, 32610 Seri Iskandar, Perak Darul Ridzuan, Malaysia. Email: mfadzil_hassan@utp.edu.my*
*[3]Centre for Foundation Studies, Universiti Teknologi PETRONAS, 32610 Seri Iskandar, Perak Darul Ridzuan, Malaysia. Email: aliza_sarlan@utp.edu.my*
*[4]CyberSecurity Malaysia, Level 7 Tower 1, Menara Cyber Axis, Jalan Impact, 63000 Cyberjaya, Selangor Darul Ehsan, Malaysia. Email: amirudin@cybersecurity.my*

***Corresponding author:** Mazni Mohamed Jakeri*
**Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, 32610 Seri Iskandar, Perak Darul Ridzuan, Malaysia. Email: maznym@yahoo.com*

***Abstract***

Adaptive security activities are a list of recommended security activities to be integrated smoothly with the software development life cycle (SDLC) to produce a secure application software. Adaptive security activities are needed due to the emergence of factors and constraints which have been determined as one of the reasons for the underutilisation of security activities implementation, especially in the earlier phase of software development process. Security activities selection models were proposed to select and recommend security activities but the models were focused on certain factors or as a solution for specific constraints, and thus the recommended security activities were not adaptive. Consequently, an adaptive security activities selection (ASAS) model was proposed by combining the factors and constraints faced by the development team in selecting security activities. The model consisted of two integrated multi-criteria decision-making (MCDM) methods, namely Analytic Network Process (ANP) and Reference Ideal Method (RIM). ANP was used to prioritise and weight the criteria while RIM was used to measure and evaluate the security activities with the value of constraints in regard to each criterion. To validate the model a case study was performed on four in-house web application development teams in the Malaysian public sector. The proposed model was able to recommend security activities in the requirement and design phase based on different constraints faced by each of the development teams. The model was adaptive due to its flexibility and ability to change and suit different evolved conditions when recommending the security activities.

***Keywords: Adaptive security activities; criteria; constraints; ANP; RIM.***

## INTRODUCTION

Software development life cycle (SDLC) is a framework which describes activities performed throughout the development process and focuses completely on functionality and features. In terms of security, there is a need to implement security throughout the entire development process (Positive Technologies, 2017; MAMPU, 2016). Security-related activities are integrated with each phase of the existing development process to set up a secure SDLC (SSDLC) (Batcheller *et al.*, 2017). For example, by integrating misuse cases in the requirement phase, threat modelling in design phase, code review in the development phase and penetration test in testing phase. The purpose of performing any security activity is to increase security posture of the SDLC artefact on which the activity is performed.

Organisations have published secure frameworks that integrate security activities in the SDLC as reference for organisations and developers who aims to reduce the number and severity of vulnerabilities in software, such as Cybersecurity Guidelines for SSDLC (CyberSecurity Malaysia, 2019), Microsoft Security Development Lifecycle (MS SDL) (Microsoft Corporation, 2010), Cigital Touchpoints (McGraw, 2006) and NIST Special Publication, SP 800-64, Revision 2 (Kissel et al., 2008). However, the implementation of security activities is influenced by many factors and constraints. Amongst these factors are security training and awareness, automated tool support, adequate development time and budget/ cost (Kanniah & Mahrin, 2016). Meanwhile, constraints are lack of security knowledge (Assal & Chiasson, 2018; Deschene, 2016), lack of experience and skills (Stephens, 2017), limited budget (Assal & Chiasson, 2018), limited development timeline (Maher, 2020)**,** insufficient human resource (Deschene, 2016), excessive workload (Assal & Chiasson, 2018), and lack of security tools (Stephens, 2017).

Previous studies showed that researchers had proposed security activities selection models by selecting the "best practices" of security activities from existing SSDLC frameworks to satisfy specific factors or as a solution for specific constraints faced by developers in selecting and recommending security activities. Factors such as cost, benefit, time, effort, and expertise were used by researchers as the main basis in ranking and selecting security activities for software development integration, either for the traditional or agile development process. For example, A. Sharma and Bawa (2020) used cost and benefit while Koc et al. (2019) used time and cost as the main factors to select and evaluate the security activities. Both A. Sharma and Bawa (2020) and Koc et al. (2019) used a survey for data collection to evaluate the listed security activities.

Several attempts have been made by using constraints in proposing frameworks to ease the integration and implementation of security activities. Mythily et al. (2019) proposed an Auto Secure Business Process (AutoSBP) system to automate security incorporation for security requirements in the design phase of existing software models as a way to reduce time and cost. To overcome the time-consuming constraints, Khamaiseh and Xu (2017) proposed a framework for constructing security test models that could automatically generate security tests. Hu et al. (2017) introduced a formal security model based on Z language to replace the manual verification of a security model due to heavy workload and reduce the cost of testing and maintenance. Dubey and Muthukrishnan (2016) proposed a platform that provided a uniform view of warnings from multiple static analysis tools as a solution for lack of immediate access to knowledge and guidance in performing static analysis. Bandi et al. (2019) proposed embedding secure programming concepts during the introductory programming courses due to lack of expertise in using secure programming practices.

So far, this model has been applied only in selecting the security activities that are limited to certain factors, such as cost, benefit, effectiveness and agility or as a solution for constraints such as the need for extra cost, time, effort, as well as lack of knowledge and expertise. Therefore, the selected security activities or solutions are limited to certain factors or constraints and force developers to refer to other models to find suitable security activities for other factors or constraints. Consequently, an adaptive security activities selection model (ASAS) is proposed by combining the factors and constraints simultaneously in recommending security activities. A flexible model is needed to measure, select, rank and recommend security activities by considering the diverse developers' requirements that consist of various factors, constraints and evolving conditions simultaneously to meet the developer's requirements. The recommended security activities must be adaptive, to change to suit

different conditions (Cambridge University Press, 2020) so that the recommended security activities can be implemented to produce a secure software.

Reference ideal method (RIM), is one of the multi-criteria decision-making (MCDM) methods which measures, evaluates, and ranks the security activities based on constraints (Cables *et al*., 2016). Additionally, it does not eliminate security activities that do not meet the constraints. Due to these advantages, RIM has been selected to measure the distance between alternatives (which refers to the security activities) and the value of constraints determined by the developers as well as classifies whether the security activities satisfy or violate the constraints. Then, the security activities are ranked according to criteria prioritisation through the analytic network process (ANP). The proposed model was used in a case study participated by four in-house web application development teams in the Malaysian public sector. The result showed that RIM was able to recommend adaptive security activities for the requirement and design phase by taking into account the security activities that did not meet the value of constraints that have been set by the development teams.

## RELATED WORK

Multi-criteria decision-making (MCDM) is a very important branch of decision-making theory. Over the past few decades, a number of MCDM methods were developed to deal with the measurement of tangible/intangible conflicting criteria and measurement of the decision alternatives with respect to each criterion (Saaty & Ergu, 2015). MCDM is referred to as a method used for scoring or ranking a finite number of alternatives by considering multiple conflicting criteria attached to the alternatives (Abdullah *et al*., 2018). MCDM is defined as a procedure to assess real-world circumstances based on various qualitative/quantitative criteria in certain/uncertain/ risky environments so that an appropriate course of action/choice/strategy/policy amongst several available options could be obtained (Zavadskas *et al*., 2014).

Preference ranking organization method for enrichment evaluations (PROMETHEE) V and V2 are two MCDM methods that evaluate constraints in decision-making. PROMETHEE V, an extension of PROMETHEE I and PROMETHEE II is used to re-evaluate the ranked alternatives by PROMETHEE I and II with the constraint to obtain compromised solutions (Rangel *et al*., 2015) by using integer linear programming (IP) (Fontana & Morais, 2011).
PROMETHEE V2, an extension to PROMETHEE V, was proposed by Mavrotas and Rozakis (2009) to give more degrees of freedom in the decision-making process. PROMETHEE V2 uses information from PROMETHEE I and bi-objective IP model instead of the single IP model used in PROMETHEE V. It is applied to evaluate the constraints and generate a Pareto optimal solution that categorises the alternatives as a green set (selected alternatives), red set (rejected alternatives) and grey set (subjected to subsequent decision phase). The Decision Maker (DM) is given alternatives in the green and grey set as a final decision.

However, both PROMETHEE V and V2 evaluate the constraints after comparison of each alternative is done by eliminating the variable/value that violates the constraints. Ideally, alternatives that do not meet the constraints should also be considered. RIM is a new MCDM method proposed by Cables *et al*. (2016) to rank security activities with the value of constraints. RIM is based on the method to obtain alternatives based accordingly on the maximum value and/or minimum value to obtain the alternatives that are nearest to the Positive Ideal solution (PIS) and as far as possible from the Negative Ideal Solution (NIS). However, one or several criteria may not need to have the maximum or minimum value. Therefore, RIM approaches enable users to evaluate alternatives without the need for ideal values of the criteria to be maximums (PIS) or minimums (NIS), but the values can be a value or any set of values between the minimum and maximum values (Cables *et al*. 2016). The integration of AHP, RIM, and Fuzzy RIM (FRIM) was used in military training aircraft selection, whereby the flight instructors defined the value of constraints for each criterion (Sánchez-Lozano & Rodríguez, 2020). Meanwhile, the AHP-RIM combination was used in web service selection whereby users were required to give the value of constraints for each criterion (Serrai et al., 2017). FRIM and RIM were used to evaluate alternatives with the value of constraints and rank the alternatives based on weight determined using AHP.

## METHODOLOGY

The proposed model comprised adoption of ANP and RIM to proactively recommend adaptive security activities in the SDLC phases based on the value of constraints provided by the development teams. ANP was used to

prioritise and weigh the criteria while the RIM was used to rank and recommend the adaptive security activities by measuring and evaluating the security activities with value of constraints provided by the development team for each conflicting criterion. Then, the weighted criteria were applied to rank the violated security activities. The top-ranked security activities were recommended as the best solutions. The details were elaborated on in the next section.

**ANP**

The ANP is one of the most complex MCDM methods, but on the other hand, it is a method that takes into account the most data about decision-making problems as compared to other MCDM methods (Kadoić, 2018). The ANP is a generalisation of analytic hierarchy process (AHP) by considering the dependence between elements of the hierarchy. Priorities are established in the same way as in the AHP by using pairwise comparisons and judgment of DM; however, it calculates weight more precisely. Many decision problems cannot be structured hierarchically because they involve the interaction and dependence of higher-level elements in a hierarchy on lower-level elements. Therefore, ANP is represented by a network rather than a hierarchy (Saaty, 2006).
The matrix manipulation proposed by Saaty and Takizawa (1986) was selected due to its simplification. DM was required to identify the degree of importance for each criterion through the pairwise comparison matrix of criteria based on Table 1. The consistency ratio (CR) was used to measure the consistency of DM judgement in performing the pairwise comparison for each criterion. To accept the judgement, CR value must be less than 0.10. However, if the CR value is more than 0.10, the judgement has to be repeated until the value is acceptable.

**TABLE 1.** *The scale*

| Intensity of importance | Definition |
|---|---|
| 1 | Equal importance |
| 3 | Moderate importance of one over another |
| 5 | Strong or essential importance |
| 7 | Very strong or demonstrated importance |
| 9 | Extreme importance |
| 2,4,6,8 | Intermediate values |
| Use reciprocals for inverse comparisons | |

**Source:** *(Saaty, 2006)*

**RIM**

RIM measures the distance of security activities from the value of constraints known as reference ideal (RI). If the evaluated security activities violate the constraints, which do not meet the RI, the function value will be less than 1. The more distance it is from the value of 1 the farther it is from the RI, and will be ranked at the very bottom, but not eliminated. The traditional MCDM ranking methods, such as TOPSIS, VIKOR, and SAW are then used to rank the security activities that satisfy the value of constraints. The steps are as follows:

**STEP 1: DEFINE THE WORK CONTEXT.**
The range, valuation matrix *X*, reference ideal (RI), and weight are defined. The SLDC phase that is given attention is the requirement and design phase; therefore, the security activities selected are from both phases.

**RANGE**
Range denotes “any interval, labels set, or a simple set of values that belongs to domain D” (Cables *et al*., 2016). In this study, range refers to the minimum and maximum values for each criterion. Those criteria were:

- Development team size (DTS),
- Development timeline (DT),
- Budget/ Cost (BC) which was categorised into:
  - Software procurement (Sw)
  - Security training (ST)
- Team workload (TW)
- Experience, skill, and knowledge (ESK)

The range for DTS, DT, Sw, and ST was derived from analysis of the questionnaire distributed to 201 officers, which consisted of the Information Technology (IT) Officer and Assistant Information Technology (IT) Officer who were responsible or involved in managing and developing the in-house web applications in the Malaysian public sector. A total of 102 questionnaires were returned, which reflected a 50.7% response rate. However, only 56 (54.9%) of the questionnaires were completed responses, while 46 (45.1%) were incomplete.

Table 2 shows the range for DTS, DT, Sw, ST, TW, and ESK. DTS denoteed the number of developers involved, which comprised the IT Officer and Assistant IT Officer. DT refers to timeline given for the requirement and design phase in the software development process. BC is the budget/cost allocated for Sw and ST. The range for DTS, DT, and BC was based on the minimum and maximum values given by respondent. The range for TW and ESK was not derived from the questionnaire. TW refers to whether the listed security activities will provide an additional workload to developers or not. Therefore, the range of TW was set to either 'No' or 'Yes'. ESK refers to the level of experience, skill, and knowledge needed to perform the listed security activities and it was based on the competencies proficiency scale, whereby 1-Basic, 2-Novice, 3-Intermediate, 4-Advanced, and 5-Expert (National Institute of Health, n.d.)

**TABLE 2.** Range for DTS, DT, BC, TW, and ESK

| Criteria | Minimum | Maximum | Range |
|---|---|---|---|
| DTS | 1 | 10 | [1,10] |
| DT: | | | |
| a) Requirement phase | 1 | 6 | [1, 6] |
| b) Design phase | 1 | 6 | [1, 6] |
| BC: | | | |
| a) Sw | 0 | 100,000 | [0, 100000] |
| b) ST | 0 | 40,000 | [0, 40000] |
| TW | - | - | No/ Yes |
| ESK | - | - | [1, 5] |

## VALUATION MATRIX *X*

The valuation matrix *X* refers to the value of each alternative in correspondence with the defined criteria (Cables *et al*., 2016). In this study, valuation matrix *X* represented the minimum requirement needed to perform the security activities for each criterion. The security activities are listed in Table 3.

**TABLE 3.** List of security activities

| Secure frameworks | Security activities | |
|---|---|---|
| | Requirement phase | Design phase |
| Cybersecurity guideline for SSDLC (CyberSecurity Malaysia, 2019) | • Sources for security requirement<br>• Data classification<br>• Use case and misuse case modeling<br>• Risk management | • Core security design consideration<br>• Additional design consideration<br>• Threat modeling |
| MS SDL-Simplified (Microsoft Corporation, 2010) | • Establish security requirements<br>• Create quality gates/ bug bars<br>• Security and privacy risk assessment | • Establish design requirements<br>• Analyze attack surface<br>• Threat modeling |
| Cigital Touchpoints (McGraw, 2006) | • Security requirements<br>• Risk analysis<br>• Abuse cases | • Risk analysis |
| NIST Special Publication, SP 800-64, Revision 2 (Kissel et al., 2008) | • Initiate security planning<br>• Categorize the information system<br>• Assess business impact<br>• Assess privacy impact<br>• Ensure the use of secure information system development processes | • Assess risk to the system<br>• Select and document security controls<br>• Design security architecture<br>• Engineer in security and develop controls<br>• Develop security documentation<br>• Conduct testing (developmental, functional, and security) |

The valuation matrix *X* was based on the score list given to five security experts and three practitioners in web application development in the Malaysian public and private sectors. They were required to provide the minimum requirement to perform any of the listed security activities for each criterion based on their experience, skill, and knowledge. They were also welcome to suggest the security activities that they have implemented in their agencies. Four security activities from the requirement phases were eliminated due to no responses given through the score list. Those security activities were: create quality gates/ bug bars, security and privacy risk assessment, data classification, and risk management.

Table 4 shows the valuation matrix *X* based on the feedback from respondents. An additional security activity was incorporated, namely security uses cases as prescribed by a security expert from CyberSecurity Malaysia. A competencies proficiency scale (1-Basic, 2-Novice, 3-Intermediate, 4-Advanced, and 5-Expert) from the National Institute of Health (n.d.) was used to set the minimum requirement needed to perform the listed security activities for ESK**.** The value of "No/Yes" for TW was used to show whether the security activity added additional workload to the developers. DTS denoted the minimum number of developers involved and comprised IT Officer and/or Assistant IT Officer. DT refers to the minimum timeline needed for each phase to perform the security activity. BC is for the minimum budget/ cost in Malaysian Ringgit (MYR) needed to provide the software (Sw) (which refers to the tools/ method/ software used) and the security training (ST). Sw for Malaysian Public Sector Information Security Risk Assessment System (MyRAM) was provided by the Malaysian Administrative Modernisation and Management Planning Unit (MAMPU) online and ST was organised by the respondent. Since most of the tools/methods/ software involved were brainstorming, the respondents did not put any value on Sw and ST. The same was for the Microsoft threat modelling tools which was also not given any value for Sw and ST.

**TABLE 4.** Valuation matrix X

| Security activities | **Tools/ methods/ software** | **ESK** | **TW** | **DTS** | **DT** | **BC** | | **Respondent** |
|---|---|---|---|---|---|---|---|---|
| | | | | | | **Sw** | **ST** | |
| a) Requirement phase | | | | | | | | |
| Security requirement, $A_1$ | Brain-storming | 4 | Y | 2 | 1 | 0 | 0 | SE |
| Risk analysis, $A_2$ | MyRAM | 3 | Y | 4 | 3 | 0 | 20,000 | SE |
| Risk analysis, $A_3$ | Brain-storming | 3 | Y | 2 | 1 | 0 | 0 | P |
| Misuse cases, $A_4$ | Brain-storming | 4 | Y | 2 | 1 | 0 | 0 | SE |
| Abuse cases, $A_5$ | Brain-storming | 4 | Y | 2 | 1 | 0 | 0 | SE |
| Security use cases, $A_6$ | Brain-storming | 4 | Y | 2 | 1 | 0 | 0 | SE |
| b) Design phase | | | | | | | | |
| Security design, $B_1$ | Brain-storming | 4 | Y | 2 | 1 | 0 | 0 | SE |
| Additional design, $B_2$ | Brain-storming | 3 | N | 2 | 1 | 0 | 0 | P |
| Attack surface reduction, $B_3$ | Brain-storming | 3 | Y | 2 | 1 | 0 | 0 | SE |
| Threat modelling, $B_4$ | Microsoft threat modelling tools | 3 | Y | 2 | 1 | 0 | 0 | SE |

Notes: SE = Security expert, P = Practitioner

## REFERENCE IDEAL (RI)

Reference Ideal (RI) refers to "an interval, labels set, or simple values that represent the maximum importance or relevance in a given range, which can be any set between the minimum and maximum values or can be a point" (Cables *et al*., 2016). The RI was used as a reference point in measuring and evaluating each alternative. In this study, the RI was the value of constraints and represented the limitation faced by the in-house development team for each criterion and the provided value must be within range.

## WEIGHT

A pairwise comparison matrix of criteria is used to determine the weight and normalise the weight of criteria, *W*. The pairwise comparisons are done in terms of which criterion dominates the other. The DM corresponds to questions such as "between ESK and TW, which one was more important in implementing security; and by how much?" The scale used for pairwise comparison is the scale by Saaty (2006), as shown in Table 1. The judgment consistency is checked by dividing the consistency index (CI) by the appropriate value in Table 5.

**TABLE 5.** Average Random Consistency

| Matrix Size | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Random Consistency | 0 | 0 | 0.58 | 0.9 | 1.12 | 1.24 | 1.32 | 1.41 | 1.45 | 1.49 |

**STEP 2: VALUATION MATRIX *X* NORMALISATION BY USING RIM.**

Valuation matrix *X* normalisation calculates the value of function *f*. It measures the distance between the listed security activities and the RI. In this study, if the value of valuation matrix *X* was less or equal to the RI, the value of function *f* was set to 1. The function of matrix *X* normalisation for ESK, DT, DTS, and BC are shown below.

If the valuation matrix $X <= RI$
$f = 1;$
Else {}

$$f(x, [A,B], [C,D]) = \begin{cases} 1 & \text{if } x \in [C,D] \\ 1 - \dfrac{d_{min}(x, [C,D])}{|A-C|} & \text{if } x \in [A,C] \quad \wedge\ A \neq C \\ 1 - \dfrac{d_{min}(x, [C,D])}{|D-B|} & \text{if } x \in [D,B] \quad \wedge\ D \neq B \end{cases} \quad (1)$$

where:

- *[A,B]* is range that belongs to a universe of discourse
- *[C,D]* represents reference ideal (RI)
- $x \in [A,B]$
- $[C,D] \in [A,B]$
- $d_{min}(x,[C,D])$ is distance of valuation matrix *X* to RI; calculated as follows: $d_{min}(x,[C,D]) = min(|x-C|,|x-D|)$

The selection of function *f* relies on the *x* value:

- The first function is selected if the *x* value is in the values of *s*, where $x \in [C, D]$.
- The second function is applied if the value of *x* is lower than the value of *s*, which is $x \in [A, C]$.
- If the value of *x* exceeds the value of *s*, which is $x \in [D, B]$, the third function is selected.
- Turning to this study, if value of *x* was lower than the value of RI, and $x \in [A, C]$, it signified that the evaluated alternative had addressed the constraint and value of function $f = 1$.

If the value of function *f* from (1) is 1, it signifies that the evaluated security activity satisfies the RI, which is $x \in [C, D]$. If the evaluated security activity violates the constraints, the value of function *f* is less than 1. The smaller the value of *f*, the more distant the security activity is from the RI. If the value of function $f = 0$, it means the evaluated security activity fully violates the RI.
The normalisation for TW was altered by adopting the truth table for NAND as shown in Table 6 below. In this table, the value of "0" represents "No" while the value of "1" represents "Yes".

**TABLE 6.** The truth table

| Inputs | | Truth table output condition |
|---|---|---|
| *x* | RI | NAND |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

The function for TW is:

$$f(x, [A,B], [C,D]) = \begin{cases} 1 & if\ x \notin [C,D] \\ 1 & if\ x \in [C,D] \quad \text{with A = C} \\ 0 & if\ x \in [C,D] \quad \text{with D = B} \end{cases} \qquad (2)$$

where:

- *[A,B]* is range that belongs to a universe of discourse
- *[C,D]* represents reference ideal
- $x \in [A,B]$
- $[C,D] \in [A,B]$

The valuation matrix *X* normalisation for TW follows Function (2). The evaluated security activities violated the RI if $x \in [C,D]$, where value of D = B.

**STEP 3: OBTAIN THE WEIGHTED NORMALISED MATRIX *Y'*.**

The value of *Y'* was obtained by multiplying the normalised valuation matrix *Y* determined from the value of function *f*, as presented in Step 2, with weight, *W* as shown on next page.

$$Y' = Y \otimes W \begin{pmatrix} y_{11}.w_1 & y_{12}.w_2 & \dots & y_{1n}.w_n \\ y_{21}.w_1 & y_{22}.w_2 & \dots & y_{2n}.w_n \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ y_{m1}.w_1 & y_{m2}.w_2 & \dots & y_{mn}.w_n \end{pmatrix}$$

**STEP 4: DETERMINE THE VARIATION TO THE NORMALISED RI FOR EACH SECURITY ACTIVITY**

The variation and the index to the normalised RI for each security activity, $I_i^+$ and $I_i^-$ are calculated by using the following equations.

$$I_i^+ = \sqrt{\sum_{j=1}^{n} (y'_{ij} - w_j)}^{\,2} \qquad I_i^- = \sqrt{\sum_{\cdot\cdot\cdot}^{n} (y'_{ij})} \qquad (3)$$

and $i = 1, 2, \dots, m$, $j = 1, 2 \dots, n$.

**STEP 5: RELATIVE INDEX CALCULATION FOR EACH SECURITY ACTIVITY**

Then, the relative index for each security activity was calculated by using the equation below.

$$R_i = \frac{I_i^-}{I_i^+ + I_i^-} \qquad (4)$$

where $0 < R_i < 1$, $i = 1, 2, \dots, m$

**STEP 6: RANK THE SECURITY ACTIVITIES.**

Security activities were ranked in descending order based on $R_i$ value. Top-ranked security activities reflect the best solutions.

**CASE STUDY**

The model was validated by four software development teams from selected Malaysian public sector agencies; Team 1, Team 2, Team 3 and Team 4. Each team was represented by an IT Officer and Assistant IT Officer, except for Team 3 which only consisted of two IT Officers. The IT Officer for Team 1 and Team 2 was the

Project Manager (PM) as well as the system analyst, while the Assistant IT Officer was the programmer. As for Team 3, one IT Officer acted as the PM, while the other IT Officer acted as the system analyst and programmer. Team 4 had two IT Officers who were the PM and system analyst, including an Assistant IT Officer as the programmer. They had 5 to 15 years of in-house experience in web application development. The teams were given a score list to prioritise and weight the criteria and a questionnaire to fill in the RI which represented the value of constraints for each criterion. Then, the RI gathered was used to measure, evaluate, and rank security activities. Table 7 shows the RI gathered from the development teams.

**TABLE 7.** Reference ideal for TW, DTS, DT, Sw, ST, and ESK

| Criteria | Reference Ideal | | | |
|---|---|---|---|---|
| | Team 1 | Team 2 | Team 3 | Team 4 |
| TW | No | Yes | Yes | Yes |
| DTS | 2 | 2 | 2 | 3 |
| DT: | | | | |
| a) Requirement | 1 | 2 | 2 | 1 |
| b) Design | 2 | 2 | 2 | 3 |
| BC: | | | | |
| • Sw | 100,000 | 100,000 | 0 | 0 |
| • ST | 40,000 | 40,000 | 0 | 0 |
| ESK: | | | | |
| a) Requirement phase | | | | |
| Security requirement, $A_1$ | 3 | 1 | 3 | 2 |
| Risk analysis, $A_2$ | 2 | 2 | 2 | 1 |
| Risk analysis, $A_3$ | 2 | 2 | 2 | 1 |
| Misuse cases, $A_4$ | 2 | 2 | 1 | 1 |
| Abuse cases, $A_5$ | 2 | 2 | 1 | 1 |
| Security use cases, $A_6$ | 2 | 2 | 1 | 1 |
| b) Design phase | | | | |
| Security design, $B_1$ | 3 | 1 | 3 | 2 |
| Additional design, $B_2$ | 3 | 1 | 4 | 3 |
| Attack surface reduction, $B_3$ | 3 | 2 | 3 | 1 |
| Threat modelling, $B_4$ | 1 | 2 | 1 | 1 |

Notes: 1=Basic, 2=Novice, 3=Intermediate, 4=Advanced, 5=Expert

Based on the above table, all development teams, except for Team 1, suffered from excessive workloads, for example, managing organisational events, tender documentation, as internal auditor, and multimedia production. The DT was less than a year and developed by a maximum of three team members. Team 2 and Team 3 had no budget allocation for Sw and ST. All teams had different levels of ESK competency for each listed security activity.

## RESULTS AND DISCUSSION

### WEIGHT DETERMINATION BY USING ANP

Table 8 shows the prioritisation and weight for each criterion made by each team. All CR values were less than 0.01. This implied consistency of weighted criteria concluded by the development teams. The criteria rank was set by weight in descending order. These ranks represented the degree of importance of the criteria that will affect the evaluation and selection of security activities by RIM.

**TABLE 8.** Interdependent weight of criteria, *W*, for each team

| Criteria | Team 1 | | Team 2 | | Team 3 | | Team 4 | |
|---|---|---|---|---|---|---|---|---|
| | R | *W* | R | *W* | R | *W* | R | *W* |
| ESK | 1 | 0.3588 | 3 | 0.2229 | 1 | 0.4083 | 2 | 0.1600 |
| TW | 3 | 0.1318 | 2 | 0.2485 | 5 | 0.0486 | 5 | 0.0345 |
| DTS | 6 | 0.0312 | 6 | 0.0392 | 4 | 0.1031 | 3 | 0.0914 |
| DT | 5 | 0.0637 | 4 | 0.1243 | 3 | 0.1441 | 6 | 0.0198 |
| Sw | 2 | 0.3456 | 5 | 0.0457 | 6 | 0.0370 | 1 | 0.6075 |
| ST | 4 | 0.0691 | 1 | 0.3196 | 2 | 0.2589 | 4 | 0.0868 |

Notes: R=Rank, *W*=Weight

## RECOMMENDATION OF ADAPTIVE SECURITY ACTIVITIES BY USING RIM

**Step 1:** In order to execute the normalisation process, each criterion should have associated a domain, *D* belonging to a universe of discourse and the following items for each criterion have been identified:

- Range as defined in Table 1.
- Valuation matrix *X* as defined in Table 4.
- Reference Ideal as defined in Table 7.
- Weight as determined from Table 8.

**Step 2**: Valuation matrix *X* normalisation is the process of calculating and measuring distance between security activity (valuation matrix *X*) and RI. Based on this value, the value of function *f* is calculated. The smaller the value of *f*, the more distant the security activity is from the RI which represents the higher the constraint encountered. Table 9 shows the range, RI, valuation matrix *X*, and normalised valuation matrix *Y* comprising the value of *f* for risk analysis, $A_2$ for Team 1 as an example.

**TABLE 9.** The range, RI, valuation matrix *X* and normalised valuation matrix *Y* for risk analysis, $A_2$ for Team 1

| Item | ESK | TW | DTS | DT | BC | |
|---|---|---|---|---|---|---|
| | | | | | Sw | ST |
| Range | [1, 5] | Yes/No | [1, 10] | [1, 6] | [0, 100000] | [0, 40000] |
| Valuation matrix *X* | 4 | Yes | 4 | 3 | 0 | 20,000 |
| RI | 3 | No | 2 | 1 | 100,000 | 20,000 |
| Normalised valuation matrix *Y* | 0.6667 | 1 | 0.75 | 0.6 | 1 | 1 |

The value of *f* calculated for ESK was 66.67%. This was because the minimum requirement needed to perform the evaluated security activity was 4 (Advanced) but RI given by the team was 3 (Intermediate). Therefore, from this value, the team knew their hindrance to implement the security activity and they needed to increase the ESK by 33.33% to fulfil the minimum requirement. The *f* value for Sw and ST was 1. This showed that the team had no hindrance in BC to implement the security activity. As for TW, the *f* value was 1, which was based on the truth table for NAND as shown in Table 6. Table 10 lists the normalised valuation matrix *Y* which refers to the calculated value of *f* for each team.

**TABLE 10.** Normalised valuation matrix Y

| Security activities | Team 1 | | | | | | Team 2 | | | | | | Team 3 | | | | | | Team 4 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ESK | TW | DTS | DT | BC | | ESK | TW | DTS | DT | BC | | ESK | TW | DTS | DT | BC | | ESK | TW | DTS | DT | BC | |
| | | | | | Sw | ST | | | | | Sw | ST | | | | | Sw | ST | | | | | Sw | ST |
| a) Requirement phase | | | | | | | | | | | | | | | | | | | | | | | | |
| $A_1$ | 0.5 | 1 | 1 | 1 | 1 | 1 | 0.2500 | 0 | 1 | 1 | 1 | 1 | 0.5 | 0 | 1 | 1 | 1 | 1 | 0.3333 | 0 | 1 | 1 | 1 | 1 |
| $A_2$ | 0.6667 | 1 | 0.75 | 0.6 | 1 | 1 | 0.6667 | 0 | 0.75 | 0.75 | 1 | 1 | 0.6667 | 0 | 0.75 | 0.75 | 1 | 0.5 | 0.5 | 0 | 0.8571 | 0.6 | 1 | 0.5 |
| $A_3$ | 0.6667 | 1 | 1 | 1 | 1 | 1 | 0.6667 | 0 | 1 | 1 | 1 | 1 | 0.6667 | 0 | 1 | 1 | 1 | 1 | 0.5 | 0 | 1 | 1 | 1 | 1 |
| $A_4$ | 0.3333 | 1 | 1 | 1 | 1 | 1 | 0.3333 | 0 | 1 | 1 | 1 | 1 | 0.25 | 0 | 1 | 1 | 1 | 1 | 0.25 | 0 | 1 | 1 | 1 | 1 |
| $A_5$ | 0.3333 | 1 | 1 | 1 | 1 | 1 | 0.3333 | 0 | 1 | 1 | 1 | 1 | 0.25 | 0 | 1 | 1 | 1 | 1 | 0.25 | 0 | 1 | 1 | 1 | 1 |
| $A_6$ | 0.3333 | 1 | 1 | 1 | 1 | 1 | 0.3333 | 0 | 1 | 1 | 1 | 1 | 0.25 | 0 | 1 | 1 | 1 | 1 | 0.25 | 0 | 1 | 1 | 1 | 1 |
| b) Design phase | | | | | | | | | | | | | | | | | | | | | | | | |
| $B_1$ | 0.5 | 1 | 1 | 1 | 1 | 1 | 0.25 | 0 | 1 | 1 | 1 | 1 | 0.5 | 0 | 1 | 1 | 1 | 1 | 0.3333 | 0 | 1 | 1 | 1 | 1 |
| $B_2$ | 1 | 1 | 1 | 1 | 1 | 1 | 0.5 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| $B_3$ | 1 | 1 | 1 | 1 | 1 | 1 | 0.6667 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0.5 | 0 | 1 | 1 | 1 | 1 |
| $B_4$ | 0.5 | 1 | 1 | 1 | 1 | 1 | 0.6667 | 0 | 1 | 1 | 1 | 1 | 0.5 | 0 | 1 | 1 | 1 | 1 | 0.5 | 0 | 1 | 1 | 1 | 1 |

In the requirement phase, all values of function *f* for ESK were less than 1, signifying that all the evaluated security activities had violated the constraints. The values of *f* for $A_4$, $A_5$, and $A_6$ were the lowest, indicating that the constraint in implementing those security activities was higher than $A_1$, $A_2$, and $A_3$. Both $A_2$ and $A_3$ had the highest and same values of function *f* for ESK as compared to the other criteria. However, the values of *f* of $A_2$ for DTS and DT for Team 1 were lower than 1, as well as TW, DTS and DT for Team 2 and ST in addition to Team 3 and Team 4. Therefore, the chances of $A_3$ being ranked at the top position for all teams appeared to be high. However, this depended on weight of criteria in evaluating and selecting the security activities by RIM.

In the design phase, $B_2$ and $B_3$ satisfied the RI for Team 1. Therefore, RIM cannot be used to rank the satisfied RI and other MCDM methods such as PROMETHEE could be used to rank those two security activities. $B_2$ satisfied the RI for Team 3 and Team 4 and was automatically ranked at the top position of security activities. The remaining security activities were evaluated and ranked in the next steps.

**Step 3, Step 4 and Step 5**: The weighted normalised matrix *Y'* was calculated by multiplying the weight from Table 8 with the normalised valuation matrix, *Y*, from Table 10. Then, the variation and index to normalised RI, $I_i^+$, and $I_i^-$ and relative index, $R_i$, for each security activity was calculated as shown in Table 11.

**TABLE 11.** The ranking pattern on security activities

| Security activities | Team 1 | | | Team 2 | | | Team 3 | | | Team 4 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $I_i^+$ | $I_i^-$ | $R_i$ | $I_i^+$ | $I_i^-$ | $R_i$ | $I_i^+$ | $I_i^-$ | $R_i$ | $I_i^+$ | $I_i^-$ | $R_i$ |
| a) Requirement phase | | | | | | | | | | | | |
| $A_1$ | 0.1794 | 0.4228 | 0.7021 | 0.2995 | 0.3526 | 0.5407 | 0.2098 | 0.3762 | 0.6419 | 0.1121 | 0.6230 | 0.8475 |
| $A_2$ | 0.1225 | 0.4481 | 0.7853 | 0.2614 | 0.3686 | 0.5851 | 0.1990 | 0.3315 | 0.6249 | 0.0985 | 0.6194 | 0.8628 |
| $A_3$ | 0.1196 | 0.4515 | 0.7906 | 0.2594 | 0.3786 | 0.5934 | 0.1445 | 0.4170 | 0.7426 | 0.0871 | 0.6259 | 0.8778 |
| $A_4$ | 0.2392 | 0.4011 | 0.6265 | 0.2895 | 0.3560 | 0.5515 | 0.3100 | 0.3220 | 0.5171 | 0.1249 | 0.6220 | 0.8328 |
| $A_5$ | 0.2392 | 0.4011 | 0.6265 | 0.2895 | 0.3560 | 0.5515 | 0.3100 | 0.3220 | 0.5171 | 0.1249 | 0.6220 | 0.8328 |
| $A_6$ | 0.2392 | 0.4011 | 0.6265 | 0.2895 | 0.3560 | 0.5515 | 0.3100 | 0.3220 | 0.5171 | 0.1249 | 0.6220 | 0.8328 |
| b) Design phase | | | | | | | | | | | | |
| $B_1$ | 0.1794 | 0.4228 | 0.7021 | 0.2995 | 0.3526 | 0.5407 | 0.2098 | 0.3762 | 0.6419 | 0.1121 | 0.6230 | 0.8475 |
| $B_2$ | - | - | - | 0.1114 | 0.4420 | 0.7987 | - | - | - | - | - | - |
| $B_3$ | - | - | - | 0.2594 | 0.3786 | 0.5934 | 0.0486 | 0.5162 | 0.9140 | 0.0871 | 0.6259 | 0.8778 |
| $B_4$ | 0.1794 | 0.4228 | 0.7021 | 0.2594 | 0.3786 | 0.5934 | 0.2098 | 0.3762 | 0.6419 | 0.0871 | 0.6259 | 0.8778 |

**Step 6**: The security activities were ranked based on the $R_i$ value for each team, as listed in Table 11. In the requirement phase, the $R_I$ value for $A_3$ was the highest. Therefore, $A_3$ was ranked at the top position of security activities for all teams, followed by $A_2$ and $A_1$ for Team 1, Team 2 and Team 4. $A_1$ was ranked in second place followed by $A_2$ for Team 3. Since $A_4$, $A_5$, and $A_6$ had the same $R_i$ value, they were ranked in ascending order based on the security activities code. Therefore, the DM had an opportunity to select which security activities to apply. In the design phase, $B_2$ for all teams and $B_3$ in addition to Team 1 met the RI. Therefore, $B_2$ was ranked at the top for the design phase for Team 2, Team 3 and Team 4. As for Team 1, other MCDM methods, such as PROMETHEE, could be applied to rank the $B_2$ and $B_3$. The other security activities had violated the RI and were ranked based on the $R_i$ value.

In Table 12, ESK contributed as a major constraint, and thus the developers ought to acquire adequate security training for the recommended top-ranked security activity for each phase. The TW was the second contributor for constraint violation, but not for Team 1 for all phases and $B_2$ for all teams. Based on the questionnaire feedback, the developers were involved in other tasks which included managing organisational events, tender documentation, as th internal auditor and multimedia production. This situation might be addressed by reducing the developer's workload; hence, increasing the opportunity for applying the security. ST may pose a huge constraint for Team 3 and Team 4 for $A_2$. DT, DTS and Sw were not listed as constraints for all security activities, except for $A_2$.

**TABLE 12.** Security activities ranking

| Security activities | Tools/ method/ software | Team 1 | | Team 2 | | Team 3 | | Team 4 | |
|---|---|---|---|---|---|---|---|---|---|
| | | R | C | R | C | R | C | R | C |
| a) Requirement phase | | | | | | | | | |
| Security requirement, $A_1$ | Brain-storming | 3 | ESK | 6 | ESK, TW | 2 | ESK, TW | 3 | ESK, TW |
| Risk analysis, $A_2$ | MyRAM | 2 | ESK, DTS, DT | 2 | ESK, DTS, DT, TW | 3 | ESK, DTS, DT, TW, ST | 2 | ESK, DTS, DT, TW, ST |
| Risk analysis, $A_3$ | Brain-storming | 1 | ESK | 1 | ESK, TW | 1 | ESK, TW | 1 | ESK, TW |
| Misuses cases, $A_4$ | Brain-storming | 4 | ESK | 3 | ESK, TW | 4 | ESK, TW | 4 | ESK, TW |
| Abuse cases, $A_5$ | Brain-storming | 5 | ESK | 4 | ESK, TW | 5 | ESK, TW | 5 | ESK, TW |
| Security use cases, $A_6$ | Brain-storming | 6 | ESK | 5 | ESK, TW | 6 | ESK, TW | 6 | ESK, TW |
| b) Design phase | | | | | | | | | |
| Security design, $B_1$ | Brain-storming | 3 | ESK | 4 | ESK, TW | 3 | ESK, TW | 4 | ESK, TW |
| Additional design, $B_2$ | Brain-storming | - | - | 1 | ESK | 1 | - | 1 | - |
| Attack surface reduction, $B_3$ | Brain-storming | - | - | 2 | ESK, TW | 2 | TW | 2 | ESK, TW |
| Threat modelling, $B_4$ | Microsoft threat modelling tools | 4 | ESK | 3 | ESK, TW | 4 | ESK, TW | 3 | ESK, TW |

Notes: R = Rank, C = Constraint

Table 13 shows the weight for each criterion if the DM do not perform the criteria prioritisation, whereby the scale for each pairwise comparison is set to 1. ESK was ranked at the top position followed by Sw> ST> DTS> DT> TW.

**TABLE 13.** Interdependent weight of criteria, W

| Criteria | Team 1 | |
|---|---|---|
| | R | *W* |
| ESK | 1 | 0.2083 |
| TW | 6 | 0.1250 |
| DTS | 4 | 0.1583 |
| DT | 5 | 0.1333 |
| Sw | 2 | 0.1875 |
| ST | 3 | 0.1875 |

Notes: R=Rank, *W*=Weight

Table 14 shows the $R_i$ value and ranking pattern for each security activity.

**TABLE 14.** The ranking pattern on security activities without criteria prioritisation

| Security activities | Team 1 | | | Team 2 | | | Team 3 | | | Team 4 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $I_i^+$ | $I_i^-$ | $R_i$ | $I_i^+$ | $I_i^-$ | $R_i$ | $I_i^+$ | $I_i^-$ | $R_i$ | $I_i^+$ | $I_i^-$ | $R_i$ |
| a) Requirement phase | | | | | | | | | | | | |
| $A_1$ | 0.1041 | 0.3737 | 0.7821 | 0.2001 | 0.3404 | 0.6298 | 0.1627 | 0.3521 | 0.6840 | 0.1869 | 0.3434 | 0.6476 |
| $A_2$ | 0.0960 | 0.3546 | 0.7869 | 0.1520 | 0.3372 | 0.6892 | 0.1786 | 0.2955 | 0.6233 | 0.1951 | 0.2937 | 0.6008 |
| $A_3$ | 0.0694 | 0.3848 | 0.8472 | 0.1430 | 0.3696 | 0.7179 | 0.1430 | 0.3639 | 0.7179 | 0.1627 | 0.3521 | 0.6840 |
| $A_4$ | 0.1389 | 0.3655 | 0.7246 | 0.1869 | 0.3434 | 0.6476 | 0.2001 | 0.3404 | 0.6298 | 0.2001 | 0.3404 | 0.6298 |
| $A_5$ | 0.1389 | 0.3655 | 0.7246 | 0.1869 | 0.3434 | 0.6476 | 0.2001 | 0.3404 | 0.6298 | 0.2001 | 0.3404 | 0.6298 |
| $A_6$ | 0.1389 | 0.3655 | 0.7246 | 0.1869 | 0.3434 | 0.6476 | 0.2001 | 0.3404 | 0.6298 | 0.2001 | 0.3404 | 0.6298 |
| b) Design phase | | | | | | | | | | | | |
| $B_1$ | 0.1041 | 0.3737 | 0.7821 | 0.2001 | 0.3404 | 0.6298 | 0.1627 | 0.3521 | 0.6840 | 0.1869 | 0.3434 | 0.6476 |
| $B_2$ | - | - | - | 0.1041 | 0.3737 | 0.7821 | - | - | - | - | - | - |
| $B_3$ | - | - | - | 0.1430 | 0.3696 | 0.7179 | 0.1250 | 0.3956 | 0.7599 | 0.1627 | 0.3521 | 0.6840 |
| $B_4$ | 0.1041 | 0.3737 | 0.7821 | 0.1430 | 0.3696 | 0.7179 | 0.1627 | 0.3521 | 0.6840 | 0.1627 | 0.3521 | 0.6840 |

The ranked security activities based on $R_i$ value with and without criteria prioritisation are shown in Table 15. In the requirement phase, $A_3$ remained the top-ranked security activity for all teams. $A_2$ made a very significant change in ranking for Team 3 and Team 4. The position dropped from second to the last in rankings and it was taken by $A_1$. This was because $A_2$ had the most constraints as compared to the other security activities and in turn, led to a low $R_i$ value. For other security activities, the value of $R_i$ was high due to the following reasons:

- The weight for all criteria was almost similar, except for ESK.
- The constraints were only limited to ESK for Team 1. The constraints for Team 2, Team 3 and Team 4 were limited to ESK and TW. Therefore, the value of $f$ for the other criteria was 1 since they met the RI. This in turn led to the high value of $R_i$ because of the high value $I_i^-$ and low value of $I_i^+$.

Due to the changes in ranking for $A_2$, the ranking of other security activities for Team 3 and Team 4 changed as well. The security activity ranking for the design phase remained the same.

**TABLE 15.** The ranking pattern on security activities with and without criteria prioritisation

| Security activities | Tools/ method/ software | Team 1 | | Team 2 | | Team 3 | | Team 4 | |
|---|---|---|---|---|---|---|---|---|---|
| | | W | WO | W | WO | W | WO | W | WO |
| a) Requirement phase | | | | | | | | | |
| Security requirement, $A_1$ | Brain-storming | 3 | 3 | 6 | 6 | 2 | 2 | 3 | 2 |
| Risk analysis, $A_2$ | MyRAM | 2 | 2 | 2 | 2 | 3 | 6 | 2 | 6 |
| Risk analysis, $A_3$ | Brain-storming | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Misuses cases, $A_4$ | Brain-storming | 4 | 4 | 3 | 3 | 4 | 3 | 4 | 3 |
| Abuse cases, $A_5$ | Brain-storming | 5 | 5 | 4 | 4 | 5 | 4 | 5 | 4 |
| Security use cases, $A_6$ | Brain-storming | 6 | 6 | 5 | 5 | 6 | 5 | 6 | 5 |
| b) Design phase | | | | | | | | | |
| Security design, $B_1$ | Brain-storming | 3 | 3 | 4 | 4 | 3 | 3 | 4 | 4 |
| Additional design, $B_2$ | Brain-storming | - | - | 1 | 1 | 1 | 1 | 1 | 1 |
| Attack surface reduction, $B_3$ | Brain-storming | - | - | 2 | 2 | 2 | 2 | 2 | 2 |
| Threat modelling, $B_4$ | Microsoft threat modeling tools | 4 | 4 | 3 | 3 | 4 | 4 | 3 | 3 |

Notes: W=With, WO=Without

## CONCLUSION

This study highlights the need for security activities evaluation with constraints in selecting and recommending adaptive security activities throughout the software development process. This is because the constraints are always associated as a hindrance to perform security activities. Therefore, the proposed ASAS model was used to calculate and measure the distance of security activities (valuation matrix, *X*) to the value of constraints (RI). The closer the distance indicates the closer it meets the constraint and can be considered by the team for deployment. The case study showed that the model was able to recommend adaptive security activities that could be changed to suit different constraints faced by the development teams. The result revealed that ESK emerged as a major obstacle in evaluating the security activities at both the requirement and design phases. Therefore, adequate security training is required, which poses a constraint to Team 3 and Team 4 due to limitation of budget allocation. Besides, TW also needs to be taken seriously as it impedes the selection of security activities. The evaluation and ranking of security activities are affected by the weight of criteria. Therefore, the team should make the right decision in determining the criteria prioritisation so that the recommended adaptive security activities can be embedded in developing a secure web application by considering all the constraints during the decision-making process.

## ACKNOWLEDGEMENT

The authors would like to thank the Public Service Department (JPA), Universiti Teknologi PETRONAS (UTP), and the Malaysian public and private sectors for supporting this research.

## REFERENCES

1. Abdullah, L., Chan, W., & Afshari, A. (2018). Application of PROMETHEE method for green supplier selection: a comparative result based on preference functions. Journal of Industrial Engineering International, 0123456789. https://doi.org/10.1007/s40092-018-0289-z
2. Assal, H., & Chiasson, S. (2018). Security in the Software Development Lifecycle. USENIX Symposium on Usable Privacy and Security (SOUPS), 281–296.
3. Bandi, A., Fellah, A., & Bondalapati, H. (2019). Embedding security concepts on introductory programming courses. The Journal of Computing Sciences in Colleges, 34(4), 78–89.
4. Batcheller, A., Fowler, S. C., Cunningham, R., Doyle, D., Jaeger, T., & Lindqvist, U. (2017). Building on the success of building security in. IEEE Security and Privacy, 15(4), 85–87. https://doi.org/10.1109/MSP.2017.3151336
5. Cables, E., Lamata, M. T., & Verdegay, J. L. (2016). RIM-Reference Ideal Method in Multicriteria Decision Making. Information Sciences, 337–338, 1–10. https://doi.org/10.1016/j.ins.2015.12.011
6. Cambridge University Press. (2020). Cambridge Dictionary. Cambridge University Press. https://dictionary.cambridge.org
7. CyberSecurity Malaysia. (2019). Cyber security guideline for secure software development life cycle (SSDLC) (pp. 1–60).
8. Deschene, M. (2016). Embracing security in all phases of the software development life cycle : A delphi study (Issue September). Capella University.
9. Dubey, A., & Muthukrishnan, D. (2016). An approach for collaborative quality assessment of software. ACM International Conference Proceeding Series, 18-20-Febr, 190–195. https://doi.org/10.1145/2856636.2856656
10. Fontana, M. E., & Morais, D. C. (2011). Selecting a portfolio of alternatives in participatory budgeting based on multicriteria method. Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics, 3179–3183. https://doi.org/10.1109/ICSMC.2011.6084149
11. Hu, X., Zhuang, Y., Cao, Z., Ye, T., & Li, M. (2017). Modeling and validation for embedded software confidentiality and integrity. Proceedings of the 2017 12th International Conference on Intelligent Systems and Knowledge Engineering, ISKE 2017, 2018-Janua, 1–6. https://doi.org/10.1109/ISKE.2017.8258789
12. Kadoić, N. (2018). Characteristics of the Analytic Network Process, a Multi-Criteria Decision-Making Method. Croatian Operational Research Review, 9(2), 235–244. https://doi.org/10.17535/crorr.2018.0018
13. Kanniah, S. L., & Mahrin, M. N. (2016). A review on factors influencing implementation of secure software development practices. International Journal of Computer and Systems Engineering, 10(8), 2882–2889. https://doi.org/doi.org/10.5281/zenodo.1127256
14. Kissel, R., Stine, K. M., Scholl, M. A., Rossman, H., Fahlsing, J., & Gulick, J. (2008). NIST SP 800-64

Rev. 2. Security Considerations in the System Development Life Cycle. Information Security. http://dl.acm.org/citation.cfm?id=2206279%5Cnpapers2://publication/uuid/D524BF13-D081-4554-AB83-6A82E77E6EC8
15. Maher, Z. A., Shah, A., Chandio, S., Mohadis, H. M., & Rahim, N. H. B. A. (2020). Challenges and limitations in secure software development adoption - A qualitative analysis in Malaysian software industry prospect. Indian Journal of Science and Technology, 13(26), 2601–2608. https://doi.org/10.17485/ijst/v13i26.848
16. MAMPU. (2016). Cyber security framework for public sector (RAKKSSA) (p. 34). Malaysian Administrative Modernisation and Management Planning Unit (MAMPU).
17. Mavrotas, G., & Rozakis, S. (2009). Extensions of the PROMETHEE method to deal with segmentation constraints. Application in a students' selection problem. Journal of Decision Systems, 18(2), 203–229. https://doi.org/10.3166/jds.18.203-229
18. McGraw, G. (2006). Software security: Building security in. Addison-Wesley Professional.
19. Microsoft Corporation. (2010). Simplified implementation of the SDL. Microsoft Corporation.
20. National Institute of Health. (n.d.). Competencies proficiency scale. Retrieved December 1, 2019, from https://hr.nih.gov/working-nih/competencies/competencies-proficiency-scale
21. Positive Technologies. (2017). Security trends & vulnerabilities reviews web application (2017).
22. Rangel, L., Gomes, L., & Resende, R. (2015). Prioritization of telecommunication projects: decision analysis using the PROMETHEE V method. E&G Economia e Gestão, 15(41), 311–332.
23. Saaty, T. L. (2006). The analytic network process. Decision Making with the Analytic Network Process (International Series in Operations Research & Management Science), 95, 1–26. https://doi.org/10.1007/0-387-33987-6_1
24. Saaty, T. L., & Ergu, D. (2015). When is a Decision-Making Method Trustworthy? Criteria for Evaluating Multi-Criteria Decision-Making Methods. International Journal of Information Technology & Decision Making, 14(06), 1171–1187. https://doi.org/10.1142/s021962201550025x
25. Saaty, T. L., & Takizawa, M. (1986). Dependence and independence: From linear hierarchies to nonlinear networks. European Journal of Operational Research, 26(2), 229–237. https://doi.org/10.1016/0377-2217(86)90184-0
26. Sánchez-Lozano, J. M., & Rodríguez, O. N. (2020). Application of Fuzzy Reference Ideal Method (FRIM) to the military advanced training aircraft selection. Applied Soft Computing Journal, 88, 106061. https://doi.org/10.1016/j.asoc.2020.106061
27. Serrai, W., Abdelli, A., Mokdad, L., & Serrai, A. (2017). Dealing with user constraints in MCDM based web service selection. Proceedings - IEEE Symposium on Computers and Communications, Pediswesa, 158–163. https://doi.org/10.1109/ISCC.2017.8024522
28. Sharma, A., & Bawa, R. K. (2020). Identification and integration of security activities for secure agile development. International Journal of Information Technology. https://doi.org/10.1007/s41870-020-00446-4
29. Stephens, J. C. (2017). Application security statistics report. The case for DevSecOps. In WhiteHat Security (Vol. 12).
30. Zavadskas, E. K., Turskis, Z., & Kildienė, S. (2014). State of art surveys of overviews on MCDM/MADM methods. Technological and Economic Development of Economy, 20(1), 165–179. https://doi.org/10.3846/20294913.2014.892037