**JOURNAL OF INFORMATION AND COMMUNICATION TECHNOLOGY**
*https://e-journal.uum.edu.my/index.php/jict*

How to cite this article:
Amelia, T., & Mohamed, R. (2024). Reprocolla: Requirements prioritisation model with collaboration perspectives based on cost-value approach. *Journal of Information and Communication Technology, 23*(2), 211-252. https://doi.org/10.32890/jict2024.23.2.3

# Reprocolla: Requirements Prioritisation Model with Collaboration Perspectives Based on Cost-Value Approach

[1]Tan Amelia & [2]Rozlina Mohamed
[1&2]Faculty of Computing,
Universiti Malaysia Pahang Al-Sultan Abdullah, Malaysia
[1]Department of Information System – Faculty
of Technology and Informatics,
Universitas Dinamika, Indonesia

*[1]meli@dinamika.ac.id
[2]rozlina@umpsa.edu.my
*Corresponding author

Received: 1/1/2024 Revised: 7/4/2024 Accepted: 15/4/2024 Published: 30/4/2024

## ABSTRACT

The process of ranking requirements in software development is made up of various criteria and numerous stakeholders, which are properly selected for the effective prioritisation of software requirements. These requirements have encountered several challenges, including lack of scalability, complexity of pairwise comparisons and biases due to the cognitive load on stakeholders. Therefore, this research aimed to investigate and find solutions to improve the software prioritisation process using the Reprocolla model. The model was built by weighing the criteria in terms of benefits, opportunities, costs, and risks (BOCR) classification, which were then calculated using the fuzzy analytic hierarchy process (FAHP) method. Furthermore, the selection of alternatives was computed using the fuzzy technique for order preference using the similarity to an ideal solution (FTOPSIS) method. The success of Reprocolla was evaluated using seven datasets based on real projects and compared with the two existing methods for prioritising requirements, FAHP and FTOPSIS. The experiment results used inferential and descriptive statistics approaches with three indicators: accuracy, time consumption, and ease of use. Based on the three indicators mentioned above, the inferential statistics showed no significant difference between the perspectives of clients and developers. Whereas, descriptive statistics found that Reprocolla is more accurate, consumes less time, and has the highest ease of use percentage. The result showed that as stakeholders' satisfaction level increases, the software development process becomes more accurate, thereby leading to a decrease in time consumption and a rise in ease of usage. The result also showed that the development of the Reprocolla tool, a collaboration between humans and machines, enhanced the effectiveness of the requirements prioritisation process.

**Keywords:** Requirements prioritisation, BOCR, perspectives, FAHP, FTOPSIS.

## INTRODUCTION

Effective prioritisation of requirements in software development is essential, because it ensures that limited resources, such as time, budget, and manpower, are allocated to meet the most critical needs and deliver maximum value to stakeholders (Digital.ai, 2022; Trimble & Webster, 2013). Collaborative communication among stakeholders is considered the best practice in this process (Gupta & Gupta, 2022b; Heikkilä et al., 2015; Schön et al., 2015), facilitating the implementation of high-value-added functionality and clear directives for development. However, one of the barriers to successful software development is the lack of clear priorities and directives (Devulapalli et al., 2016; Digital.ai, 2022).

The requirements prioritisation process comprised several critical aspects, namely absolute, such as Moscow and $100-scale (Albuga & Odeh, 2018), and relative priorities, including Cost-Value and

Analytical Hierarchy Process (AHP) (Karlsson & Ryan, 1997; Khan et al., 2016), as well as various measurement scales, typically nominal, ordinal, ratio, interval (Aurum & Wohlin, 2005). Additionally, prioritisation strategies, namely scoping and ordering play a significant role (Viswanathan et al., 2016), alongside considering perspectives from both clients and developers (Sheemar & Kour, 2017; Sufian et al., 2018), including evaluating beneficial and non-beneficial criteria (Santos et al., 2016). Common criteria for prioritising software features comprise business value, development cost, risk and time to market, which are frequently used (Amelia & Mohamed, 2022; Hujainah et al., 2018; Sher et al., 2019).

This research developed Reprocolla, a model designed for prioritising software construction requirements. It adopted the cost-value method while increasing collaboration, aimed at overcoming three significant problems in requirements prioritisation, namely scalability, reduced pairwise comparisons and stakeholder bias. Scalability ensures accurate prioritisation despite increasing numbers of requirements and criteria. Minimising pairwise comparisons simplifies processes and improves time efficiency significantly. Furthermore, user-friendly prioritisation methods influence stakeholders' engagement, reduce biases, and ensure fairness in decision-making.

The ability of the prioritisation method to handle the increasing number of requirements effectively is termed scalability (Hujainah et al., 2018, 2021). A method is scalable, assuming it remains user-friendly despite increasing requirements. Berander et al. (2006) introduced two classification procedures for categorising prioritisation methods. The first classification is based on a method that assigns stakeholder weights to each requirement, enabling the description of the respective relative importance examples, such as AHP, planning game, cumulative voting, numerical assignment, and Wieger method. The second classification is a negotiation method, where agreements are reached among subjective evaluations provided by stakeholders, such as the Win-Win model and multi-criteria Preference Analysis Requirements Negotiation (MPARN) method. However, the method in each category faces certain limitations, including accommodating increasing requirements and depicting a lack of scalability. This implies that the methods are not practical or user-friendly enough to handle increasing demands. In AHP, as the number of requirements increases, the number of comparisons also rises, calculated as n*(n

-1)/2 (Philips Achimugu et al., 2016; Ibriwesh, Ho, & Chai, 2018). For example, ten requirements would require 10*(10-1)/2= 45 comparisons, while 20 needs 20*(20-1)/2=190 comparisons. Majority of requirements prioritisation methods encounter scalability problems (Babar et al., 2015; Bukhsh et al., 2020; Frota Dos Santos et al., 2016; Gambo et al., 2018; Hujainah et al., 2021).

Reducing pairwise comparisons aims to minimise user effort and time consumption by adopting strategies such as hierarchical dependencies (Alawneh, 2017) and grouping requirements (Ibriwesh et al., 2018). In addition, the number of criteria used in requirements prioritisation directly affects the comparison. Therefore, efforts to reduce pairwise comparisons should consider the number of requirements and criteria used because feature prioritisation includes comparisons between criteria and requirements.

Stakeholders' bias occurs during the requirements prioritisation process, as these parties play a critical role in assigning value to requirements. It is essential for stakeholders to maintain transparency and avoid hidden agendas to ensure the accuracy of the assigned values. Much research focused on incorporating client and developer input in the prioritisation process (Alawneh, 2017; Gupta & Gupta, 2022a; Sheemar & Kour, 2017; Sufian et al., 2018). Therefore, the success of requirements prioritisation depends on the ability to provide accurate analysis from the respective perspectives.

Recent research, including stakeholders and multi-criteria decision-making, is mainly conceptual (AL-Ta'ani & Razali, 2013; Arshad et al., 2023; Hujainah et al., 2020; Pamučar et al., 2018; Sheemar & Kour, 2017). This research focused on empirical evaluation due to the need for the growing recognition of related assessment, particularly in comparing the effectiveness of requirements prioritisation methods with respect to accuracy, time consumption, and ease of use (Borhan et al., 2019; Hujainah et al., 2021; Khalid & Qamar, 2019). Furthermore, the proposed model was based on the cost-value method and stakeholder collaboration. The contribution of this research includes identifying essential criteria from both client and developer perspectives, grouping criteria based on BOCR (Benefits, Opportunities, Costs, Risks) and developing Reprocolla, a new model that uses cost-value method and stakeholder collaboration.

This research is structured as follows: the next section reviews the existing literature, then details phases for each phase in the proposed Reprocolla model. After that, the following section describes the experimental preparation for the conducted case studies. The subsequent section focuses on the analysis and evaluation of the results. The last two sections address the potential validity threats and conclusions and suggest future research directives.

## RELATED WORK

### Requirements Prioritisation

In requirements development, there are four main activities: elicitation, analysis, specification, and validation (Rasheed et al., 2021). According to Wiegers (2009), requirements represent the values that stakeholders must receive. The requirements specification of a system must be complete and consistent. Completeness means that all relevant user benefits and information are defined, while consistency ensures that requirements are coherent and accessible to contradictions (Sommerville, 2016). Requirements prioritisation includes systematically identifying, evaluating, and ranking software requirements based on predefined criteria. System stakeholders, such as individuals or organisations, may directly or indirectly influence system requirements, and they play an important role. Besides documents and operational systems, these parties are an essential source of needs (Pohl & Rupp, 2015). Stakeholders, such as clients, prioritise software features based on business value, while developers estimate the time needed to implement the requirements, thereby contributing various viewpoints of the system.

Requirements prioritisation comprised two main categories: the first focuses on the order of task implementation, while the second considers stakeholder interests across dimensions, such as business value, implementation cost, risk, and personal preference (Firesmith, 2004). Prioritisation typically considers three main perspectives: business, customer, and implementation. From a business perspective, it addresses financial benefits, source importance, competitive factors, and regulatory compliance. The customer perspective focuses on customer needs, user requirements, and contractual agreements. Meanwhile, the implementation perspective focuses on the logical arrangement of requirements, order of implementation, associated costs, potential costs when not implemented, and available resources (Lehtola et al., 2004).

The adopted methods aim to enhance business success by maximising value. Several methods, including Numerical Assignment (NA), AHP, Cost-Value, $100-Test, etc., are used for this purpose (Philip Achimugu et al., 2014). Despite the availability of multiple methods, practical methods for achieving requirements prioritisation are scarce due to their various strengths and weaknesses. These methods should inherently incorporate cost to retain the values that enhance business success. Prioritising requirements mainly focuses on assessing cost and value, with due attention to addressing implementation risk when necessary (Amelia & Mohamed, 2018; Ibriwesh et al., 2018; Mougouei & Powers, 2017; Rida et al., 2016; Sie & Alami, 2016).

## Cost-Value Method

Karlsson and Ryan (1997) proposed a cost-value method for prioritising requirements based on relative cost and value. In this method, the value is assessed in terms of how candidate requirements contribute to customer satisfaction, while cost represents the expenses associated with the successful implementation. Candidate requirements are identified using AHP, which calculates the relative value and implementation costs. Subsequently, the cost-value diagram serves as a conceptual map, assisting software managers in analysing and discussing candidate requirements (Sie & Alami, 2016).

Examining the cost and value aspects in requirement prioritisation depicts the shared interest of both entrepreneurs and developers. The success of prioritising requirements depends on effective collaboration between these two parties. However, obstacles arise in reducing pairwise comparisons to maintain measurability and prevent excessive time consumption (Amelia & Mohamed, 2018).

## Requirements Prioritisation Criteria

The factors or criteria used are fundamental for determining priorities. These criteria are categorised into beneficial and non-beneficial in Multi-Criteria Decision-Making (MCDM) methods. Non-beneficial and beneficial criteria should ideally have lower and higher values, respectively. Additionally, various literature classify criteria based on project constraints (AL-Ta'ani & Razali, 2013; Alkandari & Al-Shammeri, 2017; Nurdiani et al., 2016). These refer to specific parameters that impose limitations and influence expected outcomes (Thakurta, 2016).

Amelia and Mohamed (2022), conducted a literature review which categorised criteria for requirements prioritisation into beneficial and non-beneficial factors. The criteria were then grouped based on project constraints.

1) Beneficial Attributes

   a. Project Constraints (No): Business Value, Importance, Stakeholder Satisfaction, Authority, Knowledge, Strategic Considerations, Usability, Customer Input, Performance, Easy Use, Accuracy, Visibility, Sales, Marketing, Applicability, Reliability, Urgency.
   b. Project Constraints (Yes): Quality, Impact, Scalability, Trust.

2) Non-Beneficial Attributes

   a. Project Constraints (No): Effort Estimation or Size Measurement, Penalty, Learning Experience, External Change, Technical Feasibility, Uncertainties, Developers Input, Negative Value.
   b. Project Constraints (Yes): Development Cost, Risk, Time to Market, Dependencies, Availability of Resources, Schedule, Volatility, Implementation Effort, Complexity.

In the process of acquiring requirements, stakeholders are essential alongside documents and operational systems (Pohl & Rupp, 2015). Recognising relevant stakeholders is crucial in requirements engineering. This research categorises stakeholders based on diverse perspectives, as the respective viewpoints complexly influence the criteria used in the requirements prioritisation process. The perspective classification includes both the viewpoints of clients and developers.

**Stakeholders Collaborative**

Effective collaboration among stakeholders is crucial for selecting requirements that ensure high user satisfaction levels. However, as stakeholder participation increases, achieving user satisfaction becomes more challenging due to diverse perspectives. Establishing collaboration between clients and developers is a significant challenge in prioritisation technologies that can be difficult to attain (El Bakly & Darwish, 2017). The gap between clients and developers in determining priorities is influenced by the perceptions of the necessity of these requirements. Clients may struggle to assess the costs and technical difficulties associated with specific requirements, while developers do not understand which requirements are most important to clients all the time (Wiegers, 1999). Successful software systems require collaboration between clients and developers to prioritise requirements. However, most prioritisation methods lack support for effective communication between stakeholders (Gupta & Gupta, 2018).

## Weighting of Criteria

Generating a priority list requires weighting the criteria to calculate the value of the requirements. Ensuring transparent and holistic weighting enhances stakeholder satisfaction (Shukla & Auriol, 2013). The objective factors influencing the weight of the criteria require special attention. The literature review stated that weights derived from specific methods are more accurate than those based solely on expert understanding of criteria importance (Ng & Mohamed, 2022; Pamučar et al., 2018).

Weighted Sum Model (WSM) is the most frequently used method, specifically in addressing one-dimensional problems. Equation 1 is the objective function of WSM (Triantaphyllou et al., 1998):

$$A_i^{WSM} = \sum_{j=1}^{n} W_j X_{ij} \quad \text{for } i = 1,2,3,\ldots,m \tag{1}$$

Where:

n = Number of criteria

m = Number of alternatives

$W_j$ = Weighted of the importance of each criterion

$X_{ij}$ = Matrix value of X

Data normalisation is essential for decision-making methods because it ensures that the information obtained is numeric and comparable, enabling the combination into a single score for each alternative. When selecting a normalisation method, it is crucial to ensure an appropriate representation of the model's broad scale and comparability of the aggregated criteria to obtain alternative ratings.

Equation 2 and Equation 3 are used for performing linear normalisation:

$$\overline{X}_{ij} = \frac{X_{ij}}{X_j^{max}} \tag{2}$$

$$\overline{X}_{ij} = \frac{X_j^{min}}{X_{ij}} \tag{3}$$

Where:

Xij = Matrix value of X, where i = 1,2,3…, m and j = 1,2,3,…n
n = Number of criteria
m = Number of alternatives

Weighting the criteria and alternatives by decision-makers can often be ambiguous, uncertain, and subjective. The use of fuzzy numbers helps reduce these uncertainties and conflicting requirements, leading to more reliable outcomes (Nazim et al., 2022). The fuzzy logic method was calculated based on the degree of truth rather than a binary true or false value, providing a conceptual framework to address uncertainties in knowledge representation (Ruby & Balkishan, 2015).

## Fuzzy Analytic Hierarchy Process (FAHP)

AHP is a potent method for resolving complex decision-making problems. In decision-making, it is essential to identify, analyse, and compare alternatives to achieve the desired objectives (Adepoju et al., 2020). The effectiveness of this analysis directly influences the level of success. However, some drawbacks are associated with the AHP pairwise comparison method, such as its reliance on expert judgment, which leads to imprecision. Using specific numerical values for experts' preferences can be limited by inadequate information or expertise. To overcome this challenge, a fusion of fuzzy set theory with AHP, known as fuzzy AHP or FAHP, allows for accommodating subjective impressions and judgments. Constructing the FAHP model includes creating a comparison matrix, consolidating multiple assessments, evaluating consistency, and refining fuzzy weights (Liu et al., 2020).

### Fuzzy Technique for Order Preference by Similarity to an Ideal Solution (FTOPSIS)

The Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS), proposed by Hwang and Yoon (1981), is a widely recognised method in Multi-Criteria Decision Analysis (MCDA). Several research extensively explored the application in various scenarios (Abu-Shareha, 2022; Palczewski & Sałabun, 2019). TOPSIS operates on the fundamental principle that the selected alternative should be closest to the ideal solution, with the aim to maximise profits and minimise total costs (Pourjavad & Mayorga, 2016; Srisawat & Payakpate, 2016). The merits of this method lie in the simplicity, computational efficiency, and comprehensive mathematical concepts, contributing to the widespread adoption. Furthermore, the classical TOPSIS has evolved into FTOPSIS, incorporating fuzzy logic to address MCDM problems in uncertain situations (Nazim et al., 2022).

An interesting fact is the widespread adoption of the FAHP method to establish criteria weights used in the FTOPSIS method. While every practical implementation of FTOPSIS consists of different criteria and alternatives, some may be combined.

## THE PROPOSED METHOD

Selecting the criteria for the requirements prioritisation process is a complex process, which includes selecting the wrong ones, thereby complicating the process and leading to uncertainty about preferring the appropriate criteria. The literature review reported the use of many criteria, and adopted excessive number without precision for decision-makers to select the process efficiently. This research used the BOCR method developed by Saaty (2005).

The criteria outlined in the literature review were grouped into four categories: Benefits, Opportunities, Costs, and Risks, collectively referred to as BOCR. Benefits (B) and Opportunities (O) pertain to factors expected from selecting priority requirements, while Costs (C) and Risks (R) are associated with meeting these requirements. The decision tree method was used to classify the criteria into BOCR categories (Amelia & Mohamed, 2023).

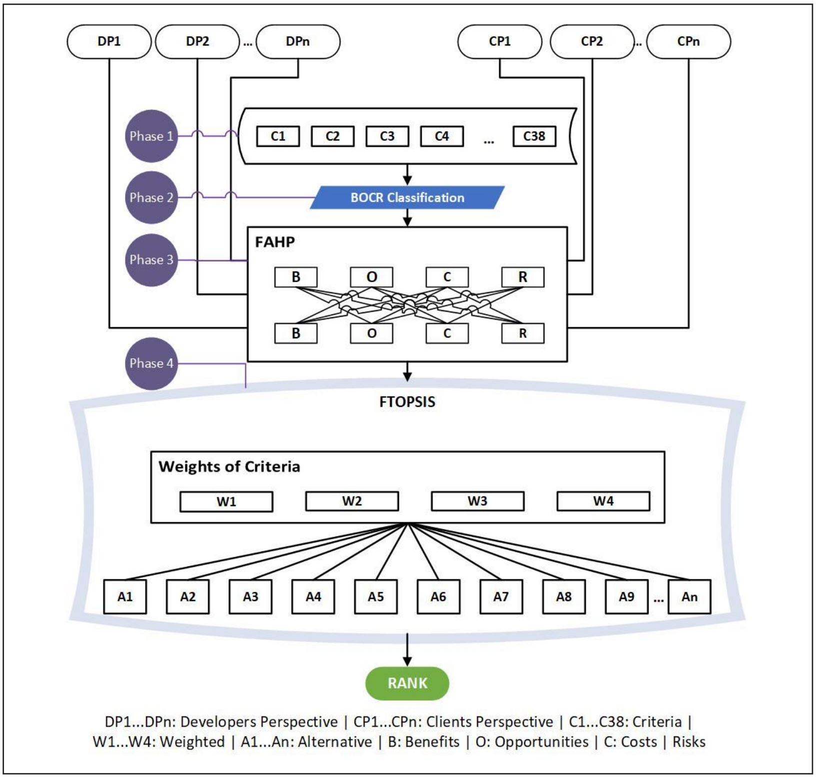
After determining the criteria, it becomes crucial to consider the stakeholders' perspectives in the requirements prioritisation process.

Collaboration between clients and developers is essential for effective prioritisation. From the perspective of clients, stakeholders comprise individuals with roles such as customers, users, top managers, and businesses. The developers' perspectives include the roles associated with software architects, analysts, designers, technicians, builders, testers, and financial representatives.

The relationship between criteria and perspective is evident in the cost-value method formulated by Karlsson and Ryan (1997). This method, which focuses on cost (non-beneficial) and value (beneficial), appeals to entrepreneurs and software developers. Considering the four BOCR merits, incorporating the perspective of clients and developers ensures that the decision-makers are well-equipped for the tasks. The phases of the proposed model are depicted in the hierarchical structure in Figure 1.

**Figure 1**

*Hierarchical Structure of the Proposed Model*



The proposed model consists of the following detailed phases:
Phase 1: Identify correo each criterion using FAHP
Phase 4: Evaluate the prioritisation using FTOPSIS

1) Identify correlation attributes

The analysis conducted based on previous research including 44 respondents (Amelia & Mohamed, 2023) shows a correlation between attributes and criteria in requirement prioritisation. From the 38 criteria identified in the literature review by Amelia and Mohammed (2023), the top ten include business value, development cost, risk, time to market, dependencies, effort estimation or size measurement, schedule, volatility, implementation effort, and stakeholder satisfaction. However, among the criteria, those selected by a high percentage of respondents are business value (27%), stakeholder satisfaction (21%) and schedule (12%).

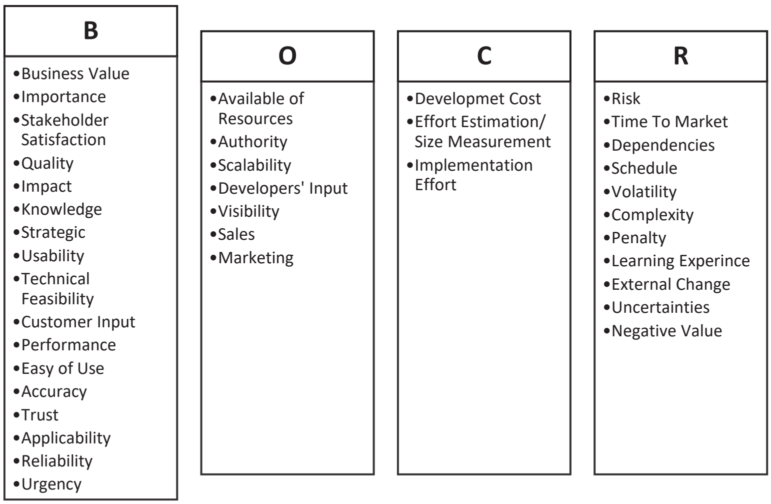
An analysis was conducted on the relationship between client and developer perspectives, as well as the correlation among criteria used in requirements prioritisation based on previous research. The results showed that perspectives of clients and developers must be consistent, while considering the viewpoints, interests, and experiences of the respective stakeholders. Certain criteria such as business value, dependencies, effort estimation, schedule, and volatility show a consistent direction with the percentage of criteria, reflecting the level of importance. However, criteria such as risk, time to market, implementation effort, development cost, and stakeholder satisfaction show an opposing direction. The correlation coefficient quantifies the strength of the relationship among the criteria used in the requirements process, all of which show a positive association, depicting the tendency to move in the same direction.

2) Classify criteria into BOCR categories

To classify the 38 criteria outlined in the literature review (Amelia & Mohamed, 2022) into BOCR categories, questionnaires and interviews were conducted with nine experts. Experts with extensive software engineering experience from both perspectives of clients and developers held positions such as program and product managers, including system analysts. Using the decision tree method, experts grouped the criteria into BOCR. The results obtained from the questionnaires and interviews confirmed that all criteria could be effectively classified into BOCR classification, as shown in Figure 2.

**Figure 2**

*The Criteria Based on BOCR*




3) Assign weights to each criterion using FAHP

Requirements prioritisation starts with calculating the weights of Benefits, Opportunities, Costs, and Risks using FAHP. This requires clients and developers conducting pairwise comparisons to assess the relative importance of attributes using fuzzy linguistic terms. The resulting weights assigned to each criterion, are then obtained from these comparisons.

Chang (1996) extended method was the most popular fuzzy AHP due to the thorough examination of each criterion. The widespread application across various real-life problems further depicts the reliability (Mangla et al., 2017; Nazim et al., 2022; Tan et al., 2014).

The following are the phases outlined in the AHP fuzzy process based on the method proposed by Chang (Prakash & Barua, 2016).

a. Calculate the fuzzified pairwise comparison matrix using Equation 4

$$\bar{\mathrm{A}} = \begin{bmatrix} 1 & \bar{\mathrm{a}}_{12} & \cdots & \bar{\mathrm{a}}_{1n} \\ \bar{a}_{21} & 1 & \cdots & \bar{\mathrm{a}}_{2n} \\ \vdots & \vdots & \ddots \cdots & 1 \\ \bar{a}_{n1} & \bar{a}_{n2} & & \end{bmatrix} = \begin{bmatrix} 1 & \bar{\mathrm{a}}_{12} & \cdots & \bar{\mathrm{a}}_{1n} \\ 1/\bar{a}_{21} & 1 & \cdots & \bar{\mathrm{a}}_{2n} \\ \vdots & \vdots & & \\ 1/\bar{a}_{1n} & 1/\bar{a}_{n2} & \ddots \cdots & 1 \end{bmatrix} \tag{4}$$

b. Calculate the fuzzy synthetic extent with respect to i[th] alternative using Equations 5-8.

$$Si = \sum_{j=1}^{n} \tilde{a}_{ij} \left[ \sum_{i=1}^{n} \sum_{j=1}^{n} \tilde{a}_{ij} \right]^{-1} \tag{5}$$

$$\tilde{A}_1 \oplus \tilde{A}_2 = (l_1, m_1, u_1) \oplus (l_2, m_2, u_2) = (l_1 + l_2, m_1 + m_2, u_1 + u_2) \tag{6}$$

$$(l_1, m_1, u_1)^{-1} = \left( \frac{1}{u_1}, \frac{1}{m_1}, \frac{1}{l_1} \right) \tag{7}$$

$$\tilde{A}_1 \otimes \tilde{A}_2 = (l_1, m_1, u_1) \otimes (l_2, m_2, u_2) = (l_1 l_2, m_1 m_2, u_1 u_2) \tag{8}$$

c. Calculate the degree of possibility for a convex fuzzy number to be greater than k using Equations 9-11

$$V(S_2 \geq S_1) = \sup_{y \geq x} [\min(\mu_{S2}(x), \mu_{S1}(y))] \tag{9}$$

$$V(S_2 \geq S_1) = \begin{cases} 1, & if\ m_2 \geq m_1 \\ 0, & if\ l_1 \geq u_2 \\ \dfrac{(l_1 - u_2)}{(m_2 - u_2) - (m_1 - l_1)} = \mu\mathrm{d}, \text{otherwise} & \end{cases} \tag{10}$$

$$V(S \geq S_1, S_2, \ldots, S_k) = \min V\,(S \geq S_i), \qquad i = 1,2, \ldots, k \tag{11}$$

d. Calculate the weight vector and then normalise the non-fuzzy using Equations 12-13

$$W' = (d'(A_1), d'(A_2), \ldots, d'(A_m))^T \tag{12}$$

$$W = (d(A_1), d(A_2), \ldots, d(A_m))^T \tag{13}$$

In a fuzzy logic system, initialisation requires defining the linguistic variables. These variable serve to describe the degrees or levels of a criterion value in both natural and artificial languages. It enables the comparisons of each criterion in a fuzzy environment. Table 1 shows the linguistic terms used to assess the weight criteria in FAHP.

**Table 1**

*Fuzzy Linguistic Terms and Correspondent Numbers for Each Criterion*

| Importance | Abbreviation | Fuzzy Number |
|---|---|---|
| Very Low | VL | (0, 0, 0.2) |
| Low | L | (0.05, 0.2, 0.35) |
| Medium Low | ML | (0.2, 0.35, 0.5) |
| Medium | ML | (0.35, 0.5, 0.65) |
| Medium High | MH | (0.5, 0.65, 0.8) |
| High | H | (0.65, 0.8, 0.95) |
| Very High | VH | (0.8, 1, 1) |

Triangular Fuzzy Number (TFN) is defined by three parameters (column): the left, middle, and right boundaries. These parameters are used to represent the membership functions of expression values. For example, in the first row (0, 0, 0.2), TFN represents a fuzzy number with the left, middle, and right boundaries of 0, 0, and 0.2, respectively. This implies that the degree of membership starts to increase from 0, reaches the maximum at 0, and then gradually declines at 0.2.

4) Evaluate the prioritisation using FTOPSIS

After obtaining the criteria weights, the prioritisation process compares alternatives with these criteria using fuzzy linguistic terms, based on FTOPSIS method. The process results in establishing the priority order for all existing alternatives. The following FTOPSIS phases guide this procedure (Kore et al., 2017; Ouma et al., 2015):

a. Decision makers assign ratings to both criteria and alternatives.
b. Fuzzy numbers are used to rate alternatives and assign criteria weights.
c. Formulation of a collected fuzzy decision matrix comprising alternative and criteria weights using Equations 14-17.

$$\bar{x}_{ij}^{k} = \left(a_{ij}^{k}, b_{ij}^{k}, c_{ij}^{k}\right) \tag{14}$$

$$w_{j}^{k} = \left(w_{j1}^{k}, w_{j2}^{k}, w_{j3}^{k}\right) \tag{15}$$

$$a_{ij} = \min_k\{a_{ij}^k\},\ b_{ij} = \frac{1}{K}\sum_{k=1}^{K} b_{ij}^k,\ c_{ij} = \max_k\{c_{ij}^k\} \tag{16}$$

$$w_{j1} = \min_k\{wj\ k1\},\ w_{j2} = \frac{1}{K}\sum_{k=1}^{K} wj\ k2,\ w_{j3} = \max_k\{wj\ k3\} \tag{17}$$

d. Compute fuzzy decision matrix using Equations 18-19.

$$\begin{array}{cc} & \begin{array}{cccc} C_1 & C_2 & \dots & C_n \end{array} \\ \bar{D} = \begin{array}{c} A_1 \\ A_2 \\ \vdots \\ A_m \end{array} & \begin{bmatrix} \bar{x}_{11} & \bar{x}_{12} & \dots & \bar{x}_{1n} \\ \bar{x}_{21} & \bar{x}_{22} & \dots & \bar{x}_{2n} \\ \vdots & \vdots & \ddots \dots & \vdots \\ \bar{x}_{m1} & \bar{x}_{m2} & & \bar{x}_{mn} \end{bmatrix} \end{array} \tag{18}$$

$$\bar{W} = (\bar{w}_1, \bar{w}_2, \dots, \bar{w}_n) \tag{19}$$

e. Normalise fuzzy decision matrix and compute the weighted normalised matrix using Equations 20-23.

$$\bar{R} = [\bar{r}_{ij}]_{mxn}, i = 1, 2, \dots, m; j = 1, 2, \dots, n \tag{20}$$

$$\bar{r}_{ij} = (a_{ij}/c_j^*, b_{ij}/c_j^*, c_{ij}/c_j^*) \text{ and } c_j^* = \max_i c_{ij} \text{ (benefit criteria)} \tag{21}$$

$$\bar{r}_{ij} = (\bar{a}_j/c_{ij}, \bar{a}_j/b_{ij}, \bar{a}_j/a_{ij}) \text{ and } \bar{a}_j = \min_i a_{ij} \text{ (cost criteria)} \tag{22}$$

$$\bar{P} = [\bar{p}_{ij}] \text{ where } \bar{p}_{ij} = \bar{r}_{ij} \text{ x } \bar{w}_j \tag{23}$$

f. Compute fuzzy positive ideal solution (FPIS) and fuzzy negative ideal solution (FNIS) using Equations 24-25

$$A^+ = (p_1^+, p_2^+, \dots, p_n^+) \text{ where}$$
$$p_j^+ = \max_i\{p_{ij3}\}, i = 1, 2, \dots, m; j = 1, 2, \dots, n \tag{24}$$

$$A^- = (p_1^-, p_2^-, \dots, p_n^-) \text{ where}$$
$$p_j^- = \min_i\{p_{ij1}\}, i = 1, 2, \dots, m; j = 1, 2, \dots, n \tag{25}$$

g. Compute the distance of each alternative from FPIS and FNIS using Equations 26-27.

$$d_i^+ = \sum_{j=1}^{n} d\left(\bar{p}_{ij}, p_j^+\right) = \left[\frac{1}{3}\sum_{j=1}^{n}\left(\bar{p}_{ij}, p_j^+\right)^2\right]^{1/2} \quad i = 1, 2, \ldots, m \tag{26}$$

$$d_i^- = \sum_{j=1}^{n} d\left(\bar{p}_{ij}, p_j^-\right) = \left[\frac{1}{3}\sum_{j=1}^{n}\left(\bar{p}_{ij}, p_j^-\right)^2\right]^{1/2} \quad i = 1, 2, \ldots, m \tag{27}$$

h. Compute the closeness coefficient ($CC_i$) of each alternative using Equation 28.

$$CC_i = \frac{d_i^-}{d_i^- + d_i^+} = \left(1 - \frac{d_i^+}{d_i^- + d_i^+}\right), for\ i = 1, 2, \ldots, m \tag{28}$$

5) Ranking the alternatives

In the final phase, diverse alternatives were ranked based on the decreasing order of the closeness coefficient ($CC_i$). The optimal alternative was characterised by the distance from FNIS to FPIS. Table 2 shows an overview of the linguistic terms used for evaluating alternatives in FTOPSIS.

**Table 2**

*Fuzzy Linguistic Terms and Correspondent Numbers for Each Alternative*

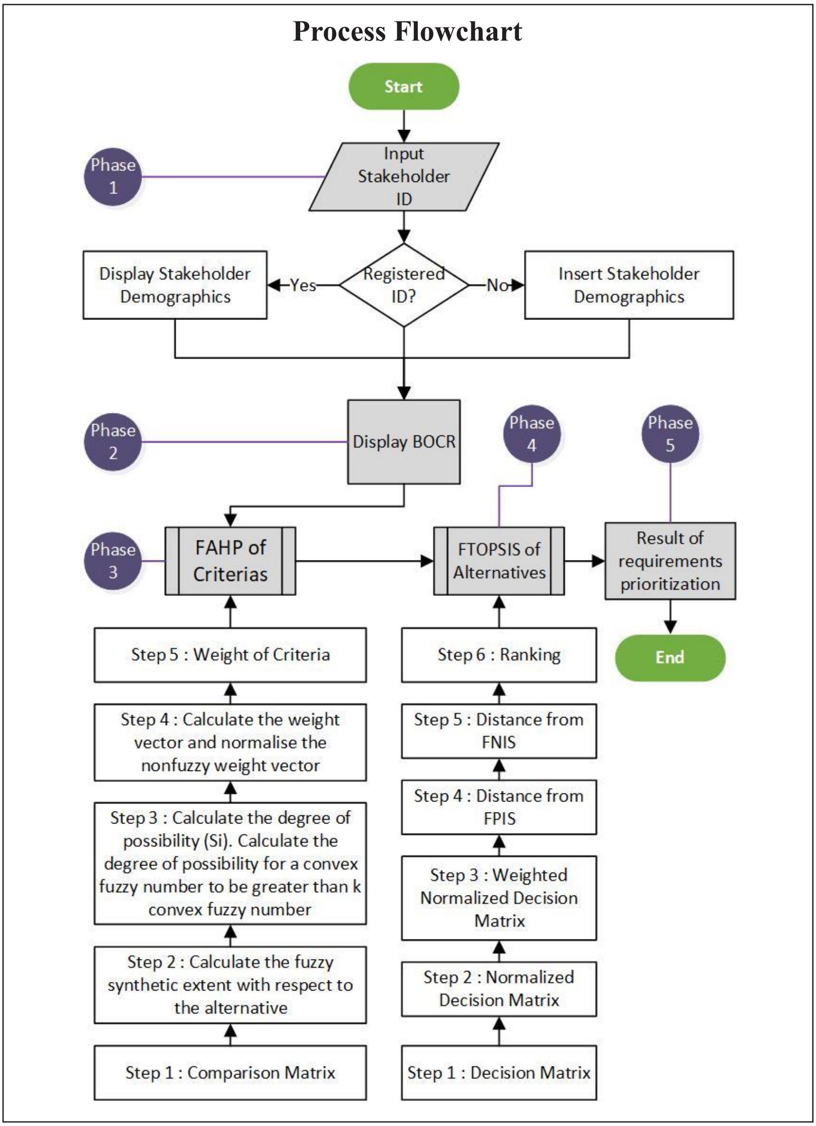
| **Importance** | **Abbreviation** | **Fuzzy Number** |
|---|---|---|
| Very Poor | VP | (0, 0, 0.2) |
| Poor | P | (0.05, 0.2, 0.35) |
| Medium Poor | MP | (0.2, 0.35, 0.5) |
| Fair | F | (0.35, 0.5, 0.65) |
| Medium Good | MG | (0.5, 0.65, 0.8) |
| Good | G | (0.65, 0.8, 0.95) |
| Very Good | VG | (0.8, 1, 1) |

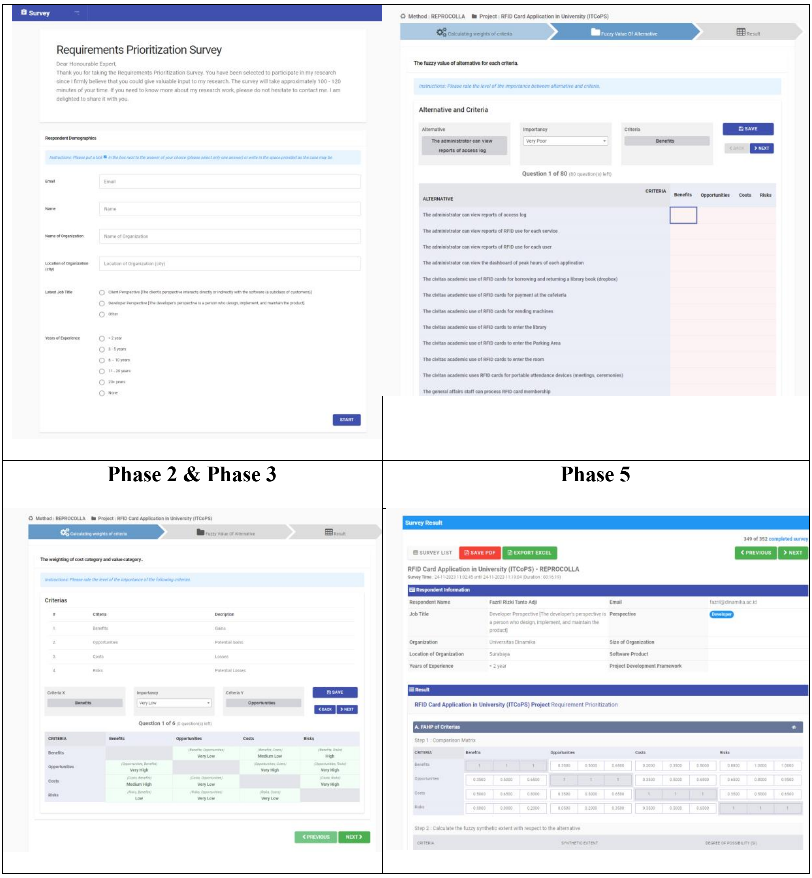
## REPROCOLLA TOOL

Reprocolla is a web-based software designed to support the automation of requirements prioritisation, aimed at improving the validation process of the proposed model. Figure 3 shows a visualisation of the system flow and interface of this tool. During the experiment, Reprocolla website was used by stakeholders for requirements prioritisation, promoting collaboration between clients and developers.

**Figure 3**

*Reprocolla Process Flowchart and Website Interface*






The implementation of the proposed model using the developed Reprocolla software consists of five distinct phases. In addition, a detailed explanation of each phase is provided below.

**Phase 1. Stakeholder Demographics**. Stakeholders enter the respective email as a unique code to access the application, name, organisation name, and location (city), Perspectives (client or developer), and years of experience.

**Phase 2. BOCR Descriptions**. The first screen, shown after logging in, provides an explanation of BOCR criteria, which ensures that stakeholders understand BOCR clearly.

**Phase 3. The Weighting of Criteria with FAHP**. Pairwise comparisons were made for each BOCR criterion by selecting the respective importance level. Subsequently, the system automatically calculates the weight for each criterion using FAHP. In the third phase, phases 1, 2, 3, 4, and 5 use formulations 4, 5 to 8, 9 to 11, 12, and 5 to 13, respectively.

**Phase 4. Calculation of Alternative with FTOPSIS**. The next phase is to assess the importance level of each alternative relative to BOCR criteria. Furthermore, the system automatically generates alternative rankings using FTOPSIS. Stakeholders are not required to input alternatives for sorting, as the system administrator has already entered them. In the fourth phase, Phases 1, 2, 3, 4, 5, and 6 use Equations 14 to 19, 20 to 22, 23, 24 to 27, 25 to 27, and 28, respectively.

**Phase 5. Results of Requirements Prioritisation**. The system showed a ranked list of requirements on the screen, saved as a pdf, and exported to Excel.

The developed semi-automated Reprocolla enables all stakeholders, including clients and developers, to easily carry out the requirements prioritisation process.

## EXPERIMENTAL SETUP

The proposed prioritisation model aims to address essential issues in ranking requirements at the start of software development, focusing on accuracy, time consumption, and ease of use. Empirical experiments, drawn from 189 data analyses, were selected to evaluate the proposed model. This research used two datasets, including real project applications developed by the Department of Information System Solutions at Universitas Dinamika and the benchmark dataset from previous investigations. The first dataset includes a three-project application, namely RFID card application (ITCoPS Application), Parking Information (PARIS Application), and Knowledge Management Systems (KRESNA Application). The second dataset comprised benchmark projects, such as (i) Replacement Access, Library and ID Card (RALIC) thesis by Lim (2010), (ii) Online Car Show Room (OCSR), (iii) Hospital Management (HMS), and (iv) Restaurant Management Systems (RMS) (Babar, 2014). The proposed model was designed for medium-sized software project development, needing between 15 to 50 requirements (Hujainah et al., 2018). Therefore, 20 out of 73 specific requirements were derived from RALIC project.

Ground truth was unavailable for OCSR, HMS, and RMS projects. The case study of bespoke development, which adhered to the software criteria with a minimum of 15 requirements, was available for clients, and comprehensive project documentation was used. The specific number of requirements from each project is shown in Table 3.

**Table 3**

*List of Number of Requirements*

| # | Project | Number of Requirements |
|---|---|---|
| 1 | ITCoPS Application | 20 |
| 2 | PARIS Application | 20 |
| 3 | KRESNA Application | 20 |
| 4 | RALIC Application | 20 |
| 5 | OCSR Application | 16 |
| 6 | HMS Application | 22 |
| 7 | RMS Application | 17 |

This experiment used a repeated measure design, incorporating counterbalancing methods to minimise confounding variables. Counterbalancing is a method used to control the effects of interfering variables in a design when the same subject is exposed to multiple conditions, treatments, or stimuli. The experiment included two respondent groups, namely clients and developers, comprising various software development roles such as product owner, system analyst, programmer, and operator. There was a total of 87 respondents, with 55 clients and 32 developers. The following are the step-by-step process of the experiment.

a. Respondents participated in the pre-test, which consisted of a questionnaire survey. Each respondent filled out the demographic questions, followed by inquiries related to requirements prioritisation.
b. Based on the counterbalancing design scenario, 87 respondents alternated between the existing and proposed models. Each respondent sequence of case studies included both similarities and differences.
c. Respondents participated in the post-test, which included the use of a survey questionnaire. Each respondent was assigned a specific requirements prioritisation model to use. After completing the

prioritisation task, questions related to requirements prioritisation in the post-test survey form were answered.

After obtaining the experimental results, the model is validated using statistical analysis. Further information regarding the model validation will be provided in the next section.

## RESULT ANALYSIS

This research evaluated the performance of Reprocolla by comparing it with existing requirements prioritisation methods, namely FAHP and FTOPSIS. Accuracy was measured by comparing these methods with the ground truth data. Time consumption was determined by measuring the processing time required for each method to complete the requirements prioritisation process. Additionally, ease of use was evaluated by comparing the opinions of respondents on certain factors such as easy to understand, use, and learn with a Likert scale for agreement measurement. Respondents provided feedback through pre and post-test questionnaires, using a five-point Likert scale the following values (1) strongly disagree/very dissatisfied, (2) disagree/dissatisfied, (3) neutral/unsure, (4) agree/satisfied, and (5) strongly agree/very satisfied. Experimental data were processed using inferential and descriptive statistics methods.

### Result of Accuracy

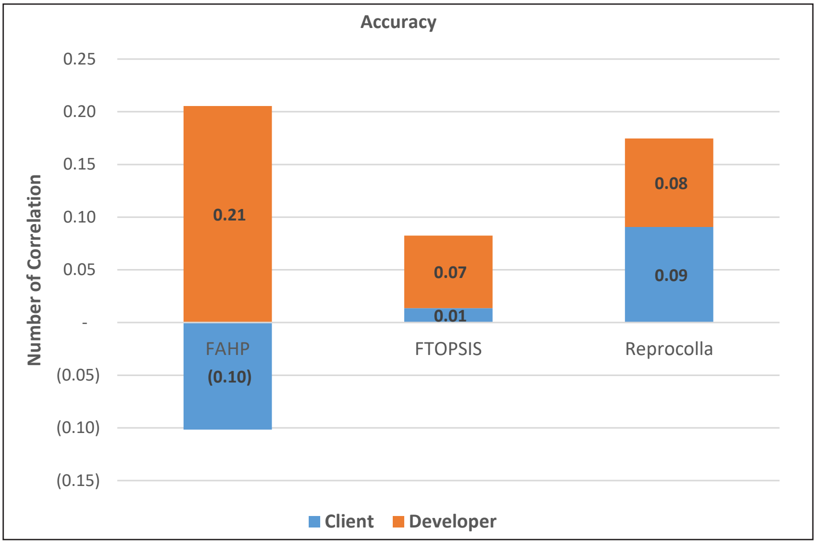
Accuracy was measured in two ways: firstly, it was calculated based on the correlation of priority results among respondents, and secondly, by satisfaction levels of respondents with the outcome obtained.

1) Based on Correlation

Figure 4 shows the accuracy results, calculated using the Pearson Correlation Coefficient, which compares the priority results obtained by each respondent to the ground truth. Reprocolla achieved the highest correlation coefficient of 0.09, followed by FAHP and FTOPSIS at 0.05 and 0.04, respectively. In this context, the measurement of accuracy refers to the consistency of rankings. A high correlation suggests agreement among stakeholders, while a lesser one depicts discrepancies that must be addressed. Analysing the correlation coefficient based on perspectives, developers and clients had 0.12, and 0.0, respectively.

**Figure 4**

*Comparison Accuracy Based on Correlation*




To determine whether there is a significant difference in accuracy between the perspectives of clients and developers, Normality in variables was assessed using Kolmogorov-Smirnova and Shapiro-Wilk tests. Table 4 shows the P-value using Kolmogorov-Smirnov and Shapiro-Wilk tests = 0.2 and 0.406, respectively, both greater than the value α = 0.05. This depicts a normal distribution at a significance level of 5 percent, which led to the conduction of a parametric analysis, namely the t-test.

**Table 4**

*Tests of Normality for Accuracy*

| Tests of Normality | | | | | | |
|---|---|---|---|---|---|---|
| | **Kolmogorov-Smirnova** | | | **Shapiro-Wilk** | | |
| | **Statistic** | **Df** | **P-Value** | **Statistic** | **df** | **P-Value** |
| Accuracy | .067 | 120 | .200* | .988 | 120 | .406 |

The hypotheses to be tested in the t-test are as follows:

$H_0$: The accuracy of results is equal for clients and developers.

$H_1$: The accuracy of results differs between clients and developers.

**Table 5**

*T-test for Accuracy*

| | | Levene's Test for Equality of Variances | | t-test for Equality of Means | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | 95% Confidence Interval of the Difference | |
| | | F | P Value | T | df | P Value (2-tailed) | Mean Difference | Std. Error Difference | Lower | Upper |
| Accuracy | Equal variances assumed | .594 | .443 | -2.428 | 118 | .017 | -.11781 | .04851 | -.21388 | -.02174 |
| | Equal variances are not assumed. | | | -2.423 | 116.079 | .017 | -.11781 | .04861 | -.21409 | -.02153 |

Table 5 shows that the two-tailed P-value is 0.017, which is less than the value $\alpha = 0.05$. Therefore, $H_0$ was rejected, indicating that the accuracy of results differs between clients and developers. The results of the descriptive analysis are stated as follows.

**Table 6**

*Analysis Descriptive for Accuracy*

| Group Statistics | | | | | |
|---|---|---|---|---|---|
| | **Perspective** | **N** | **Mean** | **Std. Deviation** | **Std. Error Mean** |
| Accuracy | Client | 62 | .0032 | .25753 | .03271 |
| | Developer | 58 | .1210 | .27390 | .03596 |

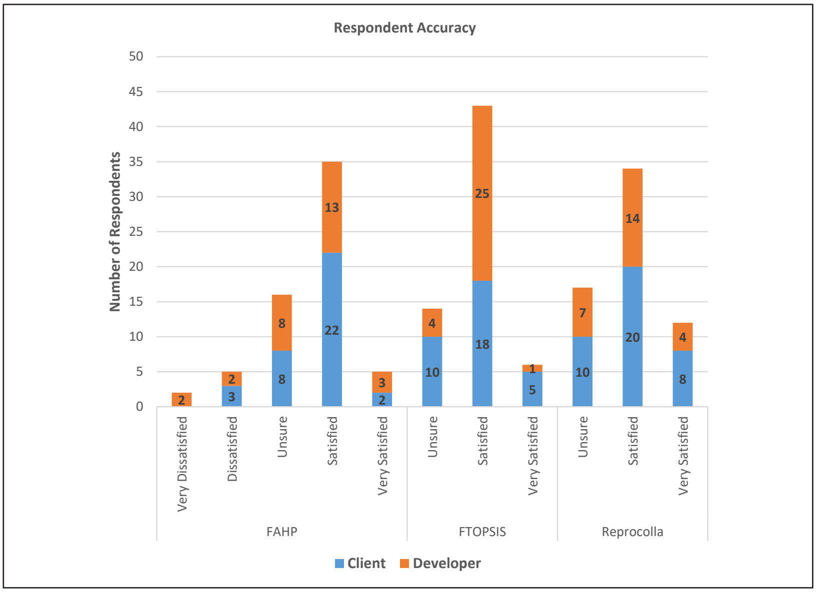
Table 6 shows that the mean correlation for clients' perspective is less (0.0032), while for developers, it is higher (0.1210). It depicts a significant difference in terms of accuracy between the two perspectives. However, in the t-test conducted for each method, only FAHP led to the rejection of the null hypothesis (p-value FAHP=0.001, FTOPSIS=0.517, Reprocolla=0.936), depicting that the accuracy of the proposed model was consistent across both perspectives of clients and developers.

2) Based on Respondent Satisfaction

In the Respondent Accuracy section, satisfaction levels were examined after reviewing each method's priority results, as shown in Figure 5. The most satisfied model was Reprocolla, followed by FTOPSIS and FAHP, with 12 (19%), 6 (10%) and 5 (8%) respondents. Regarding the satisfied options, FTOPSIS was the most frequently selected option, followed by FAHIP and Reprocolla with 43 (68%), 35 (56%) and 34 (54%) respondents, respectively. Therefore, Reprocolla attained an overall satisfaction rate of 78 percent, followed by FTOPSIS and FAHP at 77 percent and 71 percent, respectively.

**Figure 5**

*Comparison Accuracy Based on Respondent Satisfaction*




The chi-square test was used to determine whether there was a difference in satisfaction levels between clients and developers. It was selected for the suitability of ordinal scale data in a 2-sample independent or non-parametric test, which helped to determine whether there was a significant difference. The hypotheses formulated to evaluate the significance of the respondent-satisfied indicator are stated as follows:

$H_0$: The accuracy of results is equal for clients and developers
$H_1$: The accuracy of results differs between clients and developers

**Table 7**

*Chi-Square Tests for Respondent Accuracy*

| | Value | Df | P-Value |
|---|---|---|---|
| Pearson Chi-Square | 4.019a | 4 | .403 |
| Likelihood Ratio | 4.775 | 4 | .311 |
| Linear-by-Linear Association | .013 | 1 | .909 |
| N of Valid Cases | 189 | | |

Table 7 showed that the P-value for the Pearson Chi-Square statistical test was 0.403, greater than the value α= 0.05, meaning $H_0$ was accepted. Therefore, in this case, it was concluded that the accuracy of results was equal for both clients and developers.

## Result of Time-Consumption

Figure 6 shows the time required to use the model, with Reprocolla depicting the shortest duration at 18 minutes, followed by FTOPSIS and FAHP at 28 and 35 minutes, respectively. The graphic box plots in Figure 7 show that FTOPSIS has a shorter distribution of values than Reprocolla and FAHP. However, the median value of the Reprocolla dataset is the lowest (14 minutes) compared to FTOPSIS (22 minutes) and FAHP (35 minutes), depicting that it takes less time to complete on average.

**Figure 6**

*Comparison Time-Consumption*

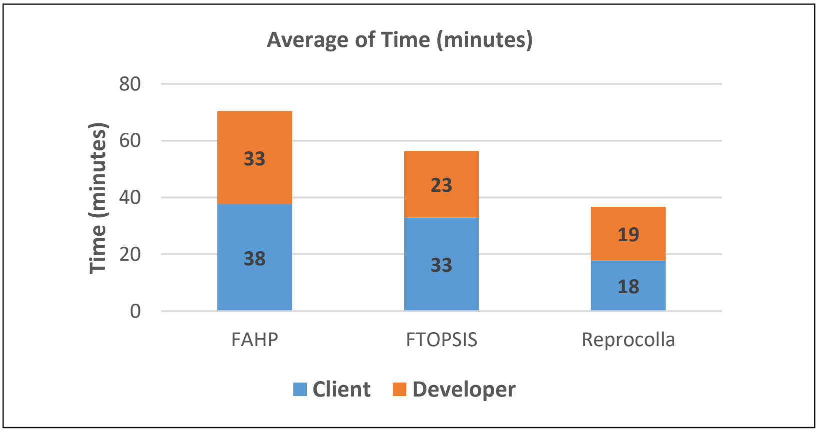
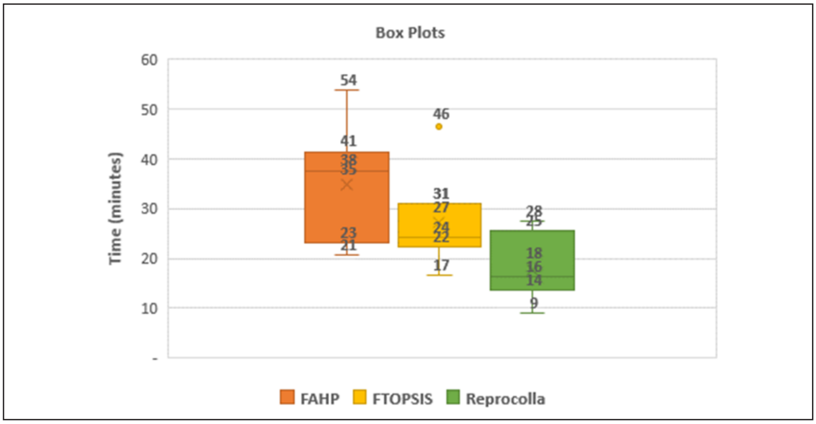


**Figure 7**

*Box Plots of Time Consumption*




To determine whether there is a significant difference in the time required to complete priority tasks between client and developer perspectives, normality in variables was examined using both Kolmogorov-Smirnova and Shapiro-Wilk tests. The resulting P-value from either test, as shown in Table 8, is 0.0001, depicting a value smaller than the $\alpha = 0.05$. At a significance level of 5%, it was concluded that the data did not follow a normal distribution. Therefore, a non-parametric difference analysis, namely the Mann-Whitney U Test, was conducted.

**Table 8**

*Tests of Normality for Time-Consumption*

| | **Tests of Normality** | | | | | |
|---|---|---|---|---|---|---|
| | **Kolmogorov-Smirnov[a]** | | | **Shapiro-Wilk** | | |
| | **Statistic** | **df** | **Sig.** | **Statistic** | **df** | **Sig.** |
| Time | .153 | 189 | .000 | .789 | 189 | .000 |

The hypotheses to be evaluated in the Mann-Whitney U Test are:

$H_0$: There is no significant difference between clients and developers with regard to the average actual time consumption to complete the prioritisation task.

$H_1$: There is a significant difference between clients and developers with regard to the average actual time consumption to complete the prioritisation task.

**Table 9**

*Mann-Whitney U Tests for Time-Consumption*

| Independent-Samples Mann-Whitney U Test Summary | |
|---|---|
| Total N | 189 |
| Mann-Whitney U | 4266.500 |
| Wilcoxon W | 7836.500 |
| Test Statistic | 4266.500 |
| Standard Error | 373.552 |
| Standardised Test Statistic | -.384 |
| P-Value | .701 |

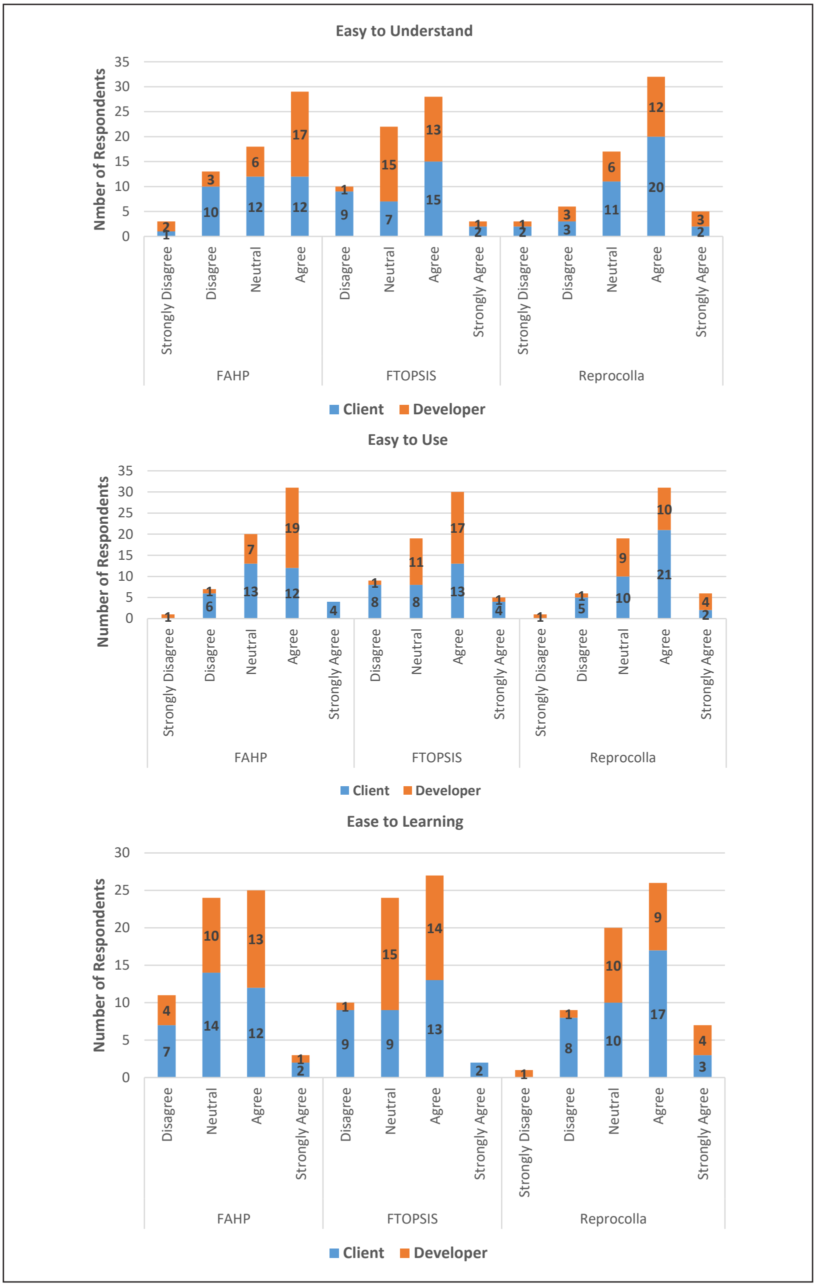
Table 9 showed that the two-tailed P-value was 0.701, greater than the value $\alpha = 0.05$, meaning $H_0$ was accepted. Therefore, it was concluded that there was no significant difference between clients and developers in the average actual time consumption to complete the prioritisation task.

**Result of Ease of Use**

Figure 8 shows three aspects associated with the ease of use for each model. Firstly, regarding ease of understanding, Reprocolla model had the highest number of respondents who selected strongly agree and agree (37, 59%), followed by FTOPSIS (31, 49%) and FAHP (29, 46%). Secondly, regarding the ease of use, Reprocolla had the highest number of respondents who strongly agreed and agreed (37, 59%), followed closely by FTOPSIS and FAHP, with 35 (56%) and 55%. Thirdly, concerning easy-to-learn, Reprocolla had the highest number of respondents (33, 52%) who strongly agreed and agreed, followed by FTOPSIS (31, 49%) and FAHP (28, 45%). Generally, Reprocolla was considered the easiest to use, followed by FTOPSIS and FAHP.

**Figure 8**

*Comparison Ease of Use*



The overall ease of use results are then calculated using a weighting value of 1 to 5 (ranging from strongly disagree to strongly agree). As shown in Table 10, Reprocolla obtained the highest average of 3.50, followed by FTOPSIS and FAHP at 3.43 and 3.32, respectively.

**Table 10**

*Average Score for Ease of Use*

| Method | Easy to Understand | Easy to Use | Ease to Learning | Average |
|---|---|---|---|---|
| FAHP | 3.16 | 3.48 | 3.32 | **3.32** |
| FTOPSIS | 3.38 | 3.49 | 3.41 | **3.43** |
| Reprocolla | 3.48 | 3.56 | 3.46 | **3.50** |

The chi-square test was used to determine whether there was a difference between clients and developers on ease of use. This evaluation, selected for suitability with ordinal scale data in a 2-sample independent test, is non-parametric in nature. The formulated hypotheses to evaluate the significance of the ease of use indicator are stated as follows:

1) Easy to Understand

$H_0$easy-to-understand: There is no significant difference between clients and developers in terms of easy to understand.

$H_1$easy-to-understand: There is a significant difference between clients and developers in terms of easy to understand.

**Table 11**

*Chi-Square Tests for Easy-to-Understand*

| | Value | Df | P-Value |
|---|---|---|---|
| Pearson Chi-Square | .338a | 4 | .987 |
| Likelihood Ratio | .339 | 4 | .987 |
| Linear-by-Linear Association | .000 | 1 | .986 |
| N of Valid Cases | 189 | | |

Table 11 showed that the P-value for the Pearson Chi-Square statistical test was 0.987, greater than the value $\alpha = 0.05$, and $H_0$ was accepted.

Therefore, no significant difference existed between clients and developers in terms of ease of understanding.

2) Easy to Use

$H_0$easy-to-use: There is no significant difference between clients and developers regarding ease of use.
$H_1$easy-to-use: There is a significant difference between clients and developers regarding ease of use.

**Table 12**

*Chi-Square Tests for Easy-to-Use*

| | **Value** | **df** | **P-Value** |
|---|---|---|---|
| Pearson Chi-Square | 5.001a | 4 | .287 |
| Likelihood Ratio | 5.735 | 4 | .220 |
| Linear-by-Linear Association | .447 | 1 | .504 |
| N of Valid Cases | 189 | | |

Table 12 showed that the P-value for the Pearson Chi-Square statistical test was 0.287, greater than the value of $\alpha = 0.05$, and $H_0$ was accepted. Therefore, no significant difference existed between clients and developers in terms of ease of use.

3) Easy to Learn

$H_0$easy-to-learn: There is no significant difference between clients and developers regarding easy-to-learn.
$H_1$easy-to-learn: There is a significant difference between clients and developers regarding easy-to-learn.

**Table 13**

*Chi-Square Tests for Easy-to-Learn*

| | **Value** | **df** | **P-Value** |
|---|---|---|---|
| Pearson Chi-Square | 4.775a | 4 | .311 |
| Likelihood Ratio | 5.153 | 4 | .272 |
| Linear-by-Linear Association | 1.180 | 1 | .277 |
| N of Valid Cases | 189 | | |

Table 13 showed that the P-value for the Pearson Chi-Square statistical test was 0.311, greater than the value $\alpha = 0.05$, and $H_0$ was accepted. Therefore, no significant difference existed between clients and developers in terms of ease of learning.

This research identified several challenges associated with validity and limitations, and among the most relevant threats to internal validity are the number of respondents, low concentration and potential selection bias. However, with 89 respondents and 189 generated experimental data, the sample size was sufficient for this experiment. The use of a prioritisation tool reduced time consumption, with an average completion time of 27 minutes per method, ensuring respondents were not exhausted. Moreover, the counterbalancing design, which determined the order of treatment, was implemented to minimise the risk of decision-making bias due to previous evaluations.

A pilot experiment was conducted to reduce potential threats to construct validity and to refine measurement instruments. Furthermore, clear descriptions were provided in the tool to guide users through each procedure. The experiments were monitored to ensure accurate completion of the questionnaire and user input. Conclusion Validity evaluates the accuracy and support of inferences derived from gathered data. This research used inferential and descriptive statistics to reduce the threat to conclusion validity.

External validity refers to generalising research results to the broader population and different settings. In this research, a potential threat to external validity arises from the type of respondents selected and the dataset used for testing. The knowledge and experience of the respondents could influence the results when determining the importance level of alternatives based on prioritisation criteria. Respondents included software developers, system information students, and product owners with relevant expertise. To ensure replicability across different contexts, this research used seven datasets from three real project applications and four benchmark information.

## CONCLUSION AND FUTURE WORK

In conclusion, the proposed model addressed scalability issues, reduced the need for pairwise comparisons, and minimised stakeholder bias. The phases in implementing the model included collecting correlation attributes, classifying criteria in BOCR, weighing criteria using FAHP, and evaluating through FTOPSIS. After synthesising literature reviews and survey findings, it was evident that both clients and developers frequently prioritised three criteria, namely business value, stakeholder satisfaction, and schedule. These criteria were identified as having significant potentials or crucial alternatives for inclusion in the requirements prioritisation process. The classification of all criteria under BOCR framework effectively conformed with cost-value method, and beneficial criteria such as Benefits and Opportunities including non-beneficial namely Costs and Risks.

The accuracy indicator of ground truth was assessed using Pearson Correlation and t-test, which showed differences in accuracy results between clients and developers. Only FAHP had different result accuracy based on the perspectives of clients and developers, while FTOPSIS and Reprocolla obtained equal results. In terms of the average accuracy based on ground truth, Reprocolla obtained the highest value, followed by FAHP and FTOPSIS. The accuracy level of the questionnaire using Chi-Square showed no disparity between clients and developers. However, Reprocolla had the highest percentage for the average level of respondent satisfaction with the results obtained. The Mann-Whitney U Test was used for the time-consumption indicator, showing no significant difference between clients and developers. The average results (in minutes) of Reprocolla were \lower than FAHP and FTOPSIS. For the last indicator, there was no significant difference between clients and developers in Chi-Square ease of use. However, Reprocolla achieved the highest percentage of ease level.

The proposed model provided valuable insights by considering the significance of the criteria and enhancing stakeholders' collaboration for prioritisation in the solution, which was essential for making strategic decisions about prioritising requirements. Diversifying stakeholder representation using structured decision-making methods and implementing transparent and accountable prioritisation processes was accommodated in the proposed model.

Reducing user intervention in requirements prioritisation was a challenging but promising area for future work, as it aimed to simplify and automate the process while preserving or even enhancing the quality of prioritisation outcomes. Further work should include improving a reference model, refining it to capture stakeholders needs, objectives, and constraints, as well as enhancing the effectiveness in guiding the prioritisation process. In addition, issues such as changes and the increasing number of requirements continued to be faced by software developers and remained open for future research.

## ACKNOWLEDGEMENT

The authors are grateful to The Center for Research and Community Service of Universitas Dinamika for supporting this research.

## REFERENCES

Abu-Shareha, A. A. (2022). TOPSIS-based regression algorithms evaluation. *Journal of Information and Communication Technology*, *21*(4), 513–547. https://doi.org/10.32890/jict2022.21.4.3

Achimugu, Philip, Selamat, A., Ibrahim, R., & Mahrin, M. N. (2014). A systematic literature review of software requirements prioritisation research. *Information and Software Technology*, *56*(6), 568–585. https://doi.org/10.1016/j.infsof.2014.02.001

Achimugu, Philips, Selamat, A., & Ibrahim, R. (2016). ReproTizer: A fully implemented software requirements prioritisation tool. *Transactions on Computational Collective Intelligence XXII*, 80–105. https://doi.org/10.1007/978-3-662-49619-0_5

Adepoju, S. A., Oyefolahan, I. O., Abdullahi, M. B., & Mohammed, A. A. (2020). Multi-criteria decision making based approaches in website quality and usability evaluation: A systematic review. *Journal of Information and Communication Technology*, *19*(3), 399–436. https://doi.org/10.32890/jict2020.19.3.5

AL-Ta'ani, R. H., & Razali, R. (2013). Prioritising requirements in agile development: A conceptual framework. *Procedia Technology*, *11*, 733–739. https://doi.org/10.1016/j.protcy.2013.12.252

Alawneh, L. (2017). Requirements prioritisation using hierarchical dependencies. In *Information Technology - New Generations* (pp. 459–464). https://doi.org/10.1007/978-3-319-54978-1_59

Albuga, S., & Odeh, Y. (2018). Towards prioritising software business requirements in startups. *2018 8th International Conference on Computer Science and Information Technology (CSIT)*, 257–265. https://doi.org/10.1109/csit.2018.8486216

Alkandari, M., & Al-Shammeri, A. (2017). Enhancing the process of requirements prioritisation in agile software development - A proposed model. *Journal of Software*, *12*(6), 439–453. https://doi.org/10.17706/jsw.12.6.439-453

Amelia, T., & Mohamed, R. (2022). A review: Requirements prioritisation criteria within collaboration perspective. *Pertanika Journal of Science and Technology*, *31*(1), 161–185. https://doi.org/10.47836/pjst.31.1.11

Amelia, T., & Mohamed, R. (2023). A decision tree approach based on BOCR for minimising criteria in requirements prioritisation. *Indonesian Journal of Electrical Engineering and Computer Science*, *32*(2), 1094–1104. https://doi.org/10.11591/ijeecs.v32.i2.pp1094-1104

Amelia, T., & Mohamed, R. B. (2018). Review on cost-value approach for requirements prioritisation techniques. *2018 5th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, 310–314. https://doi.org/10.1109/icitacee.2018.8576908

Arshad, H., Shaheen, S., Khan, J. A., Anwar, M. S., Aurangzeb, K., & Alhussein, M. (2023). A novel hybrid requirement's prioritisation approach based on critical software project factors. *Cognition, Technology and Work*, *25*(2–3), 305–324. https://doi.org/10.1007/s10111-023-00729-3

Aurum, A., & Wohlin, C. (Eds). (2005). *Engineering and managing software requirements*. Springer-Verlag. https://doi.org/10.1007/3-540-28244-0

Babar, M. I. (2014). *Framework for stakeholder quantification and requirements prioritisation for value-based software development*. Universiti Teknologi Malaysia.

Babar, M. I., Ghazali, M., Jawawi, D. N. A., Shamsuddin, S. M., & Ibrahim, N. (2015). PHandler: An expert system for a scalable software requirements prioritisation process. *Knowledge-Based Systems*, *84*, 179–202. https://doi.org/10.1016/j.knosys.2015.04.010

Berander, P., Khan, K. A., & Lehtola, L. (2006). Towards a research framework on requirements prioritisation. *Sixth Conference on Software Engineering Research and Practice in Sweden (SERPS'06)*.

Borhan, N. H., Zulzalil, H., Hassan, S., & Ali, N. M. (2019). Requirements prioritisation techniques focusing on agile software development: A systematic literature review.

*International Journal of Scientific and Technology Research*, *8*(11), 2118–2125.

Bukhsh, F. A., Bukhsh, Z. A., & Daneva, M. (2020). A systematic literature review on requirement prioritisation techniques and their empirical evaluation. *Computer Standards and Interfaces*, *69*, 103389. https://doi.org/10.1016/j.csi.2019.103389

Chang, D.-Y. (1996). Applications of the extent analysis method on fuzzy AHP. *European Journal of Operational Research*, *95*(3), 649–655. https://doi.org/10.1016/0377-2217(95)00300-2

Devulapalli, S., Rao, O., & Khare, A. (2016). Requirements prioritisation: Parameters of relevance -- An empirical study across 3 datasets. *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*. https://doi.org/10.1145/2905055.2905340

Digital.ai. (2022). State of agile report. In *Annual Report for the State of Agile* (Vol. 16). https://info.digital.ai/rs/981-LQX-968/images/AR-SA-2022-16th-Annual-State-Of-Agile-Report.pdf

El Bakly, A. H., & Darwish, N. R. (2017). A fuzzy approach for Wieger's method to rank priorities in requirement engineering. *CIIT*, *November*.

Firesmith, D. (2004). Prioritising requirements. *Journal of Object Technology*, *3*(8), 35–47. https://doi.org/10.5381/jot.2004.3.8.c4

Frota Dos Santos, J. R., Albuquerque, A. B., & Pinheiro, P. R. (2016). Requirements prioritisation in market-driven software: A survey based on large numbers of stakeholders and requirements. *2016 10th International Conference on the Quality of Information and Communications Technology (QUATIC)*. https://doi.org/10.1109/QUATIC.2016.020

Gambo, I. P., Ikono, R. N., Achimugu, P. O., & Iroju, O. G. (2018). A ranking model for software requirements prioritisation during requirements engineering: A case study. *International Journal of Computer Science and Information Security (IJCSIS)*, *16*(4), 255–268.

Gupta, A., & Gupta, C. (2018). A collaborative approach for improvisation and refinement of requirement prioritisation process. *Journal of Information Technology Research*, *11*(2), 128–149. https://doi.org/10.4018/JITR.2018040108

Gupta, A., & Gupta, C. (2022a). A novel collaborative requirement prioritisation approach to handle priority vagueness and inter-relationships. *Journal of King Saud University - Computer* *and Information Sciences*, *34*(5), 2288–2297. https://doi.org/10.1016/j.jksuci.2019.12.002

Gupta, A., & Gupta, C. (2022b). CDBR: A semi-automated collaborative execute-before-after dependency-based requirement prioritisation approach. *Journal of King Saud University - Computer and Information Sciences*, *34*(2), 421–432. https://doi.org/10.1016/j.jksuci.2018.10.004

Heikkilä, V. T., Paasivaara, M., Rautiainen, K., Lassenius, C., Toivola, T., & Järvinen, J. (2015). Operational release planning in large-scale scrum with multiple stakeholders - A longitudinal case study at F-secure corporation. *Information and Software Technology*, *57*, 116–140. https://doi.org/10.1016/j.infsof.2014.09.005

Hujainah, F., Bakar, R. A., Alhroob, E., Al-haimi, B., & Nasser, A. B. (2020). Interrelated elements in formulating an efficient requirements prioritisation technique: Review. *2020 10th IEEE International Conference on Control System, Computing and Engineering*, 97–101. https://doi.org/10.1109/ICCSCE50387.2020.9204955

Hujainah, F., Bakar, R. B. A., Abdulgabber, M. A., & Zamli, K. Z. (2018). Software requirements prioritisation: A systematic literature review on significance, stakeholders, techniques and challenges. *IEEE Access*, *6*, 71497–71523. https://doi.org/10.1109/ACCESS.2018.2881755

Hujainah, F., Bakar, R. B. A., Nasser, A. B., Al-haimi, B., & Zamli, K. Z. (2021). SRPTackle: A semi-automated requirements prioritisation technique for scalable requirements of software system projects. *Information and Software Technology*, *131*, 106501. https://doi.org/10.1016/j.infsof.2020.106501

Hwang, C.-L., & Yoon, K. (1981). *Multiple attribute decision making: Methods and applications a state-of-the-art survey*. Springer-Verlag Berlin Heidelberg.

Ibriwesh, I., Ho, S.-B., & Chai, I. (2018). Overcoming scalability issues in Analytic Hierarchy Process with ReDCCahp: An empirical investigation. *Arabian Journal for Science and Engineering*, *43*(12), 7995–8011. https://doi.org/10.1007/s13369-018-3283-2

Ibriwesh, I., Ho, S.-B., Chai, I., & Tan, C.-H. (2018). Prioritising solution-oriented software requirements using the multiple perspective prioritisation technique algorithm: An empirical investigation. *Concurrent Engineering: Research and Applications*, *27*(1), 68–79. https://doi.org/10.1177/1063293X18808559

Karlsson, J., & Ryan, K. (1997). A cost-value approach for prioritising requirements. *IEEE Software*, *14*(5), 67–74. https://doi.org/10.1109/52.605933

Khalid, Z., & Qamar, U. (2019). Weight and cluster based test case prioritisation technique. *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, 1013–1022. https://doi.org/10.1109/IEMCON.2019.8936202

Khan, S. U. R., Lee, S. P., Dabbagh, M., Tahir, M., Khan, M., & Arif, M. (2016). RePizer: A framework for prioritisation of software requirements. *Frontiers of Information Technology & Electronic Engineering*, *17*(8), 750–765. https://doi.org/10.1631/FITEE.1500162

Kore, N. B., Ravi, K., & Patil, S. B. (2017). A simplified description of FUZZY TOPSIS method for multi criteria decision making. *International Research Journal of Engineering and Technology (IRJET)*, *4*(5), 2047–2050.

Lehtola, L., Kauppinen, M., & Kujala, S. (2004). Requirements prioritisation challenges in practice. In *Lecture Notes in Computer Science* (pp. 497–508). https://doi.org/10.1007/978-3-540-24659-6_36

Lim, S. L. (2010). *Social networks and collaborative filtering for large-scale requirements elicitation* [Doctoral dissertation, University of New South Wales]. http://discovery.ucl.ac.uk/1329883/

Liu, Y., Eckert, C. M., & Earl, C. (2020). A review of fuzzy AHP methods for decision-making with subjective judgements. *Expert Systems with Applications*, *161*, 113738. https://doi.org/10.1016/j.eswa.2020.113738

Mangla, S. K., Govindan, K., & Luthra, S. (2017). Prioritising the barriers to achieve sustainable consumption and production trends in supply chains using fuzzy Analytical Hierarchy Process. *Journal of Cleaner Production*, *151*, 509–525. https://doi.org/10.1016/j.jclepro.2017.02.099

Mougouei, D., & Powers, D. M. W. (2017). Modeling and selection of interdependent software requirements using fuzzy graphs. *International Journal of Fuzzy Systems*, *19*(6), 1812–1828. https://doi.org/10.1007/s40815-017-0364-4

Nazim, M., Wali Mohammad, C., & Sadiq, M. (2022). A comparison between fuzzy AHP and fuzzy TOPSIS methods to software requirements selection. *Alexandria Engineering Journal*, *61*(12), 10851–10870. https://doi.org/10.1016/j.aej.2022.04.005

Ng, S. Y., & Mohamed, R. (2022). Feasibility study on using MCDM for E-Voting. *International Journal of Software Engineering and Computer Systems*, *8*(2), 1–9. https://doi.org/10.15282/ijsecs.8.2.2022.1.0098

Nurdiani, I., Börstler, J., & Fricker, S. A. (2016). The impacts of agile and lean practices on project constraints: A tertiary study. *Journal of Systems and Software*, *119*, 162–183. https://doi.org/10.1016/j.jss.2016.06.043

Ouma, Y. O., Opudo, J., & Nyambenya, S. (2015). Comparison of fuzzy AHP and fuzzy TOPSIS for road pavement maintenance prioritisation: Methodological exposition and case study. *Advances in Civil Engineering*, *2015*, 1–17. https://doi.org/10.1155/2015/140189

Palczewski, K., & Sałabun, W. (2019). The fuzzy TOPSIS applications in the last decade. *Procedia Computer Science*, *159*, 2294–2303. https://doi.org/10.1016/j.procs.2019.09.404

Pamučar, D., Stević, Ž., & Sremac, S. (2018). A new model for determining weight coefficients of criteria in MCDM models: Full consistency method (FUCOM). *Symmetry*, *10*(9), 393. https://doi.org/10.3390/sym10090393

Pohl, K., & Rupp, C. (2015). *Requirements engineering fundamentals: A study guide for the certified professional for requirements engineering exam foundation level / IREB compliant* (2nd ed.). Santa Barbara, CA: Rocky Nook, Inc.

Pourjavad, E., & Mayorga, R. V. (2016). A hybrid approach integrating AHP and TOPSIS for sustainable end-of-life vehicle strategy evaluation under fuzzy environment. *WSEAS Transactions on Circuits and Systems*, *15*, 216–223.

Prakash, C., & Barua, M. K. (2016). A multi-criteria decision-making approach for prioritising reverse logistics adoption barriers under fuzzy environment: Case of Indian electronics industry. *Global Business Review*, *17*(5), 1107–1124. https://doi.org/10.1177/0972150916656667

Rasheed, A., Zafar, B., Shehryar, T., Aslam, N. A., Sajid, M., Ali, N., Dar, S. H., & Khalid, S. (2021). Requirement engineering challenges in agile software development. *Mathematical Problems in Engineering*, *2021*, 1–18. https://doi.org/10.1155/2021/6696695

Rida, A., Nazir, S., Tabassum, A., Sultan, Z., & Abbas, R. (2016). Role of requirements elicitation & prioritisation to optimise quality in scrum agile development. *International Journal of Advanced Computer Science and Applications*, *7*(12), 300–306. https://doi.org/10.14569/IJACSA.2016.071239

Ruby, & Balkishan. (2015). Role of fuzzy logic in requirement prioritisation. *International Journal of Innovative Research in Science, Engineering and Technology*, *4*(6), 4290–4297.

Saaty, T. L. (2005). The analytic hierarchy and analytic network processes for the measurement of intagible criteria and for decision-making. In *Multiple Criteria Decision Analysis: State of the Art Surveys. International Series in Operations Research & Management Science* (pp. 345–405). Springer, New York, NY. https://doi.org/10.1007/0-387-23081-5_9

Santos, R., Albuquerque, A., & Pinheiro, P. R. (2016). Towards the applied hybrid model in requirements prioritisation. *Procedia Computer Science*, *91*, 909–918. https://doi.org/10.1016/j.procs.2016.07.109

Schön, E. M., Escalona, M., & Thomaschewski, J. (2015). Agile values and their implementation in practice. *International Journal of Interactive Multimedia and Artificial Intelligence*, *3*(5), 61. https://doi.org/10.9781/ijimai.2015.358

Sheemar, H., & Kour, G. (2017). Enhancing user-stories prioritisation process in agile environment. *2017 International Conference on Innovations in Control, Communication and Information Systems (ICICCI)*. https://doi.org/10.1109/ICICCIS.2017.8660760

Sher, F., Jawawi, D. N. A., & Mohammad, R. (2019). Requirements prioritisation aspects quantification for value-based software developments. *Journal of Theoretical and Applied Information Technology*, *97*(14), 3969–3979.

Shukla, V., & Auriol, G. (2013). Methodology for determining stakeholders' criteria weights in systems engineering. *Proceedings of the Posters Workshop at CSD&M*, 1–12.

Sie, A., & Alami, D. (2016). Cost-value requirements prioritisation in requirements engineering. *IEEE Software*, 1–12.

Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson Education Limited.

Srisawat, C., & Payakpate, J. (2016). Comparison of MCDM methods for intercrop selection in rubber plantations. *Journal of Information and Communication Technology*, *15*(1), 165–182. https://doi.org/10.32890/jict2016.15.1.8

Sufian, M., Khan, Z., Rehman, S., & Haider Butt, W. (2018). A systematic literature review: Software requirements prioritisation techniques. *2018 International Conference on Frontiers of Information Technology (FIT)*, 35–40. https://doi.org/10.1109/FIT.2018.00014

Tan, R. R., Aviso, K. B., Huelgas, A. P., & Promentilla, M. A. B. (2014). Fuzzy AHP approach to selection problems in process engineering involving quantitative and qualitative aspects. *Process Safety and Environmental Protection*, *92*(5), 467–475. https://doi.org/10.1016/j.psep.2013.11.005

Thakurta, R. (2016). Understanding requirement prioritisation artifacts: A systematic mapping study. *Requirements Engineering*, *22*(4), 491–526. https://doi.org/10.1007/s00766-016-0253-7

Triantaphyllou, E., Shu, B., Sanchez, S. N., & Ray, T. (1998). Multi-criteria decision making: An operations research approach. *Encyclopedia of Electrical and Electronics Engineering*, *15*, 175–186.

Trimble, J., & Webster, C. (2013). From traditional, to lean, to agile development: Finding the optimal software engineering cycle. *2013 46th Hawaii International Conference on System Sciences*, 4826–4833. https://doi.org/10.1109/HICSS.2013.237

Viswanathan, A., Nair, S. R., & Krishnan, S. M. (2016). Solution model for requirement prioritisation. *International Journal of Control Theory and Applications*, *9*(15), 7489–7496.

Wiegers, K. E. (1999). First things first: Prioritising requirements. *Software Development*, *7*(9), 48–53.

Wiegers, K. E. (2009). *Software requirements* (2nd ed.). Microsoft Press.