SEMARAK ILMU PUBLISHING

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage: https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index

ISSN: 2462-1943

Stakeholder Analysis using Fuzzy Logic Operations for Integrated User Story Prioritisation Approach in Agile-Scrum Method

Noor Hazlini Borhan[1,*], Hazura Zulzalil [2], Sa'adah Hassan[2], Norhayati Mohd Ali [2], Abu Bakar Md Sultan[2], Rami Bahsoon[3]

[1] Department of Computing & Software Engineering, Universiti Malaysia Sarawak, Kota Samarahan, Sarawak, Malaysia
[2] Department of Software Engineering & Information System, Universiti Putra Malaysia, Serdang, Selangor, Malaysia
[3] School of Computer Science, University of Birmingham, United Kingdom

ARTICLE INFO

Article history:
Received 24 October 2023
Received in revised form 3 January 2024
Accepted 4 March 2024
Available online 25 June 2024

Keywords:
Requirements prioritization; Agile software development; Stakeholder; Stakeholder analysis; Requirements engineering

ABSTRACT

Effective prioritisation of user stories in Agile-Scrum projects is pivotal to ensuring optimal allocation of resources and meeting both user and system requirements. The process of prioritising user stories requires careful stakeholder analysis to determine their influence, the significance of their story, and the value it adds to the project. Common challenges, such as conflicting views, stakeholder exclusion, requirements overflow, insufficient stakeholder input, and incorrect prioritisation of user stories, are typical. This study presents an innovative approach to user story prioritisation by integrating stakeholder analysis as part of the user story prioritisation process in the Agile-Scrum method. The analysis employs fuzzy logic operations to rank stakeholders based on selected parameters, facilitating a more comprehensive understanding of their contributions and concerns. The study's application focuses on the context of an ATM system's requirements and associated user stories. Feedback was gathered from software experts well-versed in practising the Agile methodology within their respective organisations. Results from this study highlight the positive impact of stakeholder analysis during the prioritisation process. At the same time, the software experts praised the approach's ability to integrate functional and non-functional user stories during the prioritisation process. By systematically addressing stakeholder perspectives, this approach ensures a balanced consideration of diverse user needs and system requirements, ultimately enhancing the accuracy and efficacy of user story prioritisation within Agile-Scrum projects.

1. Introduction

Requirements engineering (RE) is a crucial process in the software development life cycle (SDLC) to establish users' and system requirements [1]. Stakeholder identification is acknowledged to be one of the vital inputs to the process of requirements engineering [2]. Stakeholder analysis (SA) is therefore one of the key considerations of the requirements generation process. A stakeholder is

* *Corresponding author.*
E-mail address: bnhazlini@unimas.my

https://doi.org/10.37934/araset.47.2.7693

viewed as an actor with potential input, influence or beneficiary of the system. To secure the participation of the most significant stakeholders in the project, these stakeholders should be given elevated priority to prioritise the requirements [3] since the correct prioritised requirements that meet the stakeholders' needs are vital [4]. The problem being addressed in this study is a lack of a comprehensive and systematic method for integrating stakeholder analysis into the user story prioritisation process. The absence of a structured approach often results in imbalanced prioritisation decisions, where certain stakeholder needs are overlooked, leading to suboptimal project outcomes. Therefore, there is a clear need to address these challenges by developing a robust procedure that seamlessly incorporates stakeholder analysis, utilizing methodologies such as fuzzy logic operations, to holistically assess stakeholder contributions and concerns. This study aims to bridge the gap by proposing an Integrated User Story Prioritization Approach (i-USPA) that addresses stakeholder analysis issues within the Agile-Scrum method, ultimately enhancing the accuracy and efficacy of the prioritization process.

The novelty of prioritising user stories using the i-USPA lies in its comprehensive and innovative approach to ensure that software development projects are successful, produce high-quality software, and address both functional and non-functional requirements effectively. Some of the key points that highlight the uniqueness of i-USPA are:

1. Integrated Approach: i-USPA integrates multiple techniques, such as Planning Game, Analytic Hierarchy Process (AHP), and Stakeholder Analysis, into a single prioritisation framework. This integration allows for a holistic consideration of various aspects of user stories, ensuring that they align with project goals and stakeholder needs.
2. Planning Game (PG): The incorporation of the Planning Game, which is an agile methodology for user story prioritization, emphasizes collaboration and transparency. This ensures that stakeholders are actively involved in the prioritization process, making it more likely that the final prioritization reflects the collective vision of the project.
3. Stakeholder Analysis (SA): By conducting a thorough stakeholder analysis, i-USPA minimizes the risk of biases in the prioritisation process. Understanding the varying needs and preferences of different stakeholders allows for a more balanced and representative set of priorities.
4. Concurrent Prioritisation of Functional and Non-functional User Stories: One of the most distinctive features of i-USPA is its ability to prioritise both functional and non-functional user stories concurrently. This addresses the common problem of neglecting non-functional requirements until later in the development process. By addressing non-functional requirements early, the approach ensures that quality attributes like performance, security, and usability are considered from the outset, reducing the risk of costly rework and quality issues.

Prioritising non-functional user stories early in the project life cycle demonstrates a proactive approach to managing software quality. This is especially important in ensuring that the software meets not only its functional requirements but also the critical quality attributes that often determine the success of the application. In summary, i-USPA is a novel approach to user story prioritization that addresses the limitations of traditional methods by integrating multiple techniques, involving stakeholders, and prioritizing both functional and non-functional aspects. By emphasizing the early consideration of quality attributes, it contributes to the development of high-quality software that aligns with stakeholder needs and project goals.

This study article is structured as follows: The related work is presented in section 2 after the introduction. The proposed prioritising approach is then provided in Section 3, together with stakeholder analysis. The outcomes of the study are addressed in Section 4. Section 5 brings the paper to a close.

2. Related Work

Requirement prioritisation (RP) refers to the process of selecting the optimal set of requirements from a deluge of overlapping and conflicting requests made by numerous stakeholders involved in a software development project. Prioritisation is one of the key stages in software development, because it enables programmers to make rational decisions and determine which requirements are the most important to fulfil. This has been verified by a study which found RP to have a strong working relationship with a few other crucial technical practices, including interaction with techniques for requirements engineering, requirements analysis, and general software engineering [5]. Besides, in order to ensure the success of software projects, based on the survey done by [35], there are four project success categories: quality, scope, time, and cost.

RP is an essential part of requirements management in order to select the requirements based on a set of predetermined criteria and implement them in stages [6-8]. Stakeholders [9], system functioning [10], cost [11], processing time [11,34], risk concerns [12,13], and business values [7] are a few frequent factors that decide RP. Developers and stakeholders are always involved in RP activities [14]. The two sides have different priorities and areas of interest [15]. Stakeholders emphasise necessity, needs, and corporate principles [16,17]. Developers are aware of internal limitations such dependency between functions or needs, even though they are also worried about project qualities like effort [16] and cost [17,18].

The requirements are gathered from many stakeholders prior to the RP; therefore, the first step in the RE process is to comprehend the idea of a stakeholder. Requirements that have been thoroughly documented affect the software's quality. Software quality depends on well-defined parameters. Additionally, the software may not function properly if the exact requirements are not met. The RE process must be carried out in a way that guarantees the involvement of all right stakeholders. Prioritising stakeholders is one of the most prominent difficulties. The current system of prioritising stakeholders is inadequate and does not offer a full range of options. As a result, there would still be a chance that a project would not be accepted.

Stakeholders can be classified into two basic types of interest groups [19] and major and secondary classifications. Those who are directly, significantly, or possibly impacted by an organisation's actions are considered primary. Those who are secondary are impacted secondarily or would experience an impact which would be considered less important. This classification is almost universally accepted. There is, however, no useful mechanism to ascertain the immediate and long-term effects of each stakeholder. Nevertheless, stakeholders can also be divided into three groups and prioritised as follows: crucial, namely those who are required for the business to survive; strategic, namely those linked to pertinent opportunities or threats; or environmental issues, namely those that do not fall within either of the first two categories [3].

According to the analysis of SA in RP procedures conducted by [20], only 5 out of 66 RP procedures include the SA process in their RP procedure. These methods include RUPA, PHandler, VIRP, Evolve and mathematical programming. Evolve was the first RP method to implement the AHP and execute the SA. However, the AHP implementation is considered time-consuming, as it requires pairwise comparisons, which creates a scalability problem when there are a large number of stakeholders.

The VIRP approach was presented by [21] in 2001. The SA approach is used in this technique's prioritisation process to prioritise the needs. It implements SA by having experts grade the stakeholder profiles and determine the influence value of each participant on a scale of 1 to 10, according on the significance and potential impact the needs could have on the project's performance.

Another RP technique that SA has included in its prioritisation process is the RUPA technique [22]. With the support of experts and a project manager, AHP methods were used to manually carry out the SA procedure. The mathematical programming technique was the second RP technique to consider the SA process. This technique incorporates SA into their RP process, but unfortunately does not provide a description of how SA is carried out in their process. The most recent RP technique, PHandler [23], conducts a SA process based on the VIRP technique to solve the scalability problem in RP. Like the VIRP technique, this method also implements the SA process by assigning impact values to each stakeholder involved in the demand prioritisation. As this technique performs the SA process manually, the SA method used suffers from being time-consuming and highly dependent on specialist involvement.

To implement the SA during the prioritisation process, a better prioritisation approach needs to be developed. In this study, the prioritisation process integrates FUs and NFUs simultaneously, focusing on the agile Scrum methodology. The proposed prioritisation approach also implements the hybrid methodologies (PG and AHP) to address the problem of scalability in prioritising requirements or user stories to provide high-quality software on schedule and on a limited budget.

3. The Proposed Approach

This suggested method makes it possible to include the evaluation of numerous stakeholders. Depending on their choices, it may also take into account the importance of each requirement or user story. The ten steps of the suggested method are presented in Table 1 and Figure 1.

Table 1
The basic steps of i-USPA

Step No	Description
Step 1	A product owner (PO) will identify the relevant stakeholders for the software project
Step 2	A PO specifies and assigns weights of the stakeholders based on some parameters
Step 3	PO identifies all functional user stories (FUs) & non-functional user stories (NFUs)
Step 4	Apply Planning Game (PG) to all FUs & divide them into 3 sections
Step 5	Produce Pairwise Comparison Matrixes
Step 6	Select pair FUs & NFUs, elicit & calculate the importance degree of a NFU for a given FU
Step 7	Calculate NFUs ranking with respect to all FUs using TFN and alpha cut approach
Step 8	Calculate FUs Ranking
Step 9	Aggregate prioritized lists of FUS & NFUs provided by various stakeholders to obtain final rankings
Step 10	Estimate User stories

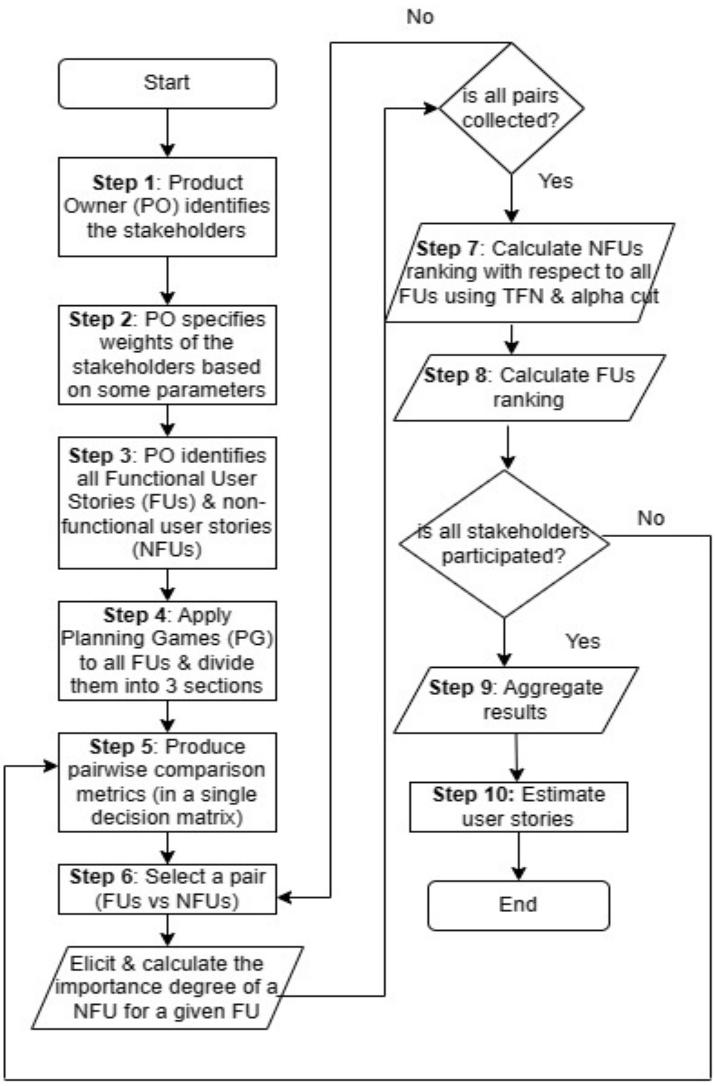

Fig. 1. The basic flow of i-USPA

To prove the concept of the proposed approach, the priority values for all the user stories involved are collected from the opinions of software practitioners involved in agile software development in their organisation.

Step 1: A critical step in the requirements elicitation process is stakeholder identification [11].

Before they can take part in the prioritisation process, the stakeholders in the system must be identified by the product owner (PO). The five stakeholders in this study who assist in prioritising both FUs and NFUs are S1, S2, S3, and S4, with another acting as PO.

Step 2: Using fuzzy logic, the PO determines the weighting of each stakeholder.

There may be some stakeholders who have the decision-making power for any software project, and priority should be given to these stakeholders to ensure the involvement of important stakeholders. The sub-steps for step 2 of the proposed i-USPA are shown in Figure 2.

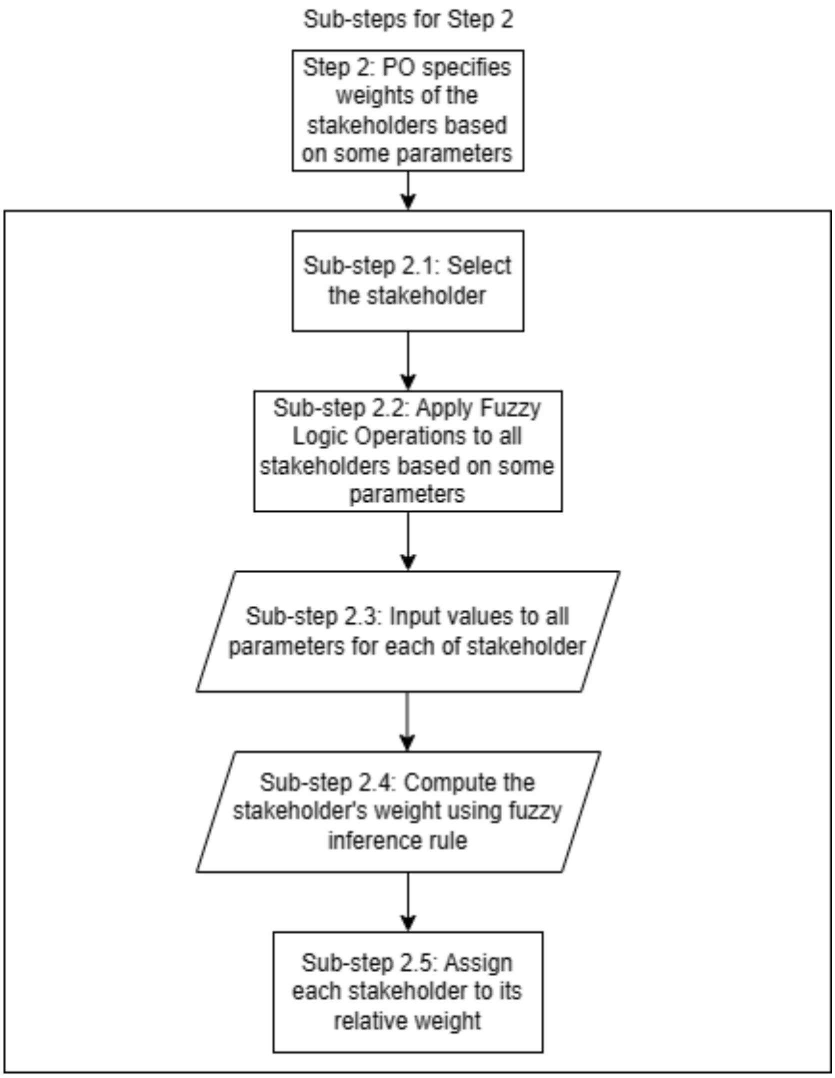

Fig. 2. Step 2 sub-steps

Sub-step 2.1: A PO will select the stakeholder.

Once the SA process is complete, the PO selects the project stakeholders and assigns them a stakeholder weighting.

Sub-step 2.2: Apply Fuzzy Logic operations to all stakeholders based on some parameters.

In this sub-step, PO ensures the SA process. There are several parameters to be considered:

a) Designation-D (Position of the individual inside the company or under the official title).
b) Experience-E (Years of Organizational Professional Experience).
c) Interpersonal Relationship- IR (This is the term used to describe the relationship between internal employees).
d) Power-P (How much power is needed to alter a proposed system's requirements?).
e) Domain knowledge- DK (Level of General Business Knowledge in the domain).
f) Technology Knowledge- TS (Average familiarity with contemporary computer architecture).

g) Influence- I (How much influence will be exerted to cause or affect a project's development?).
h) Interest - IN (How eagerly is the system responding?).

These parameters have been chosen based on the preferences of earlier research [1,25]. The values of these selected parameters were given in accordance with the stakeholder prioritisation model, as shown in Table 2 and the value matrix in Table 3 and 4.

Table 2
Model for prioritising stakeholders [24]

SLNo	No of Stakeholder
Stakeholder Name	Name of the person
Designation	Position of the person according to organization/ official title
Experience	Years of working experience in the organization
Interpersonal Relationship	Label of relationship among the internal employee
Power	How much authority exert to the system?
Domain Knowledge	State of overall business knowledge
Technology Skill	Average knowledge about modern computing system
Influence	How much influence will exert or will be exerted
Interest	How much eagerness to the system?

Table 3
Value Matrix [24]

SLNo	Designation (D)	Value	Power (P)	Value	Domain Knowledge (DM)	Value
1	Top Most	10	Absolutely High	10	Absolutely High	10
2	Top Middle	9	Very High	9	Very High	9
3	Miserable Top	8	High	8	High	8
4	Upper Middle	7	Moderate High	7	Moderate High	7
5	Middle	6	Medium	6	Medium	6
6	Lower Middle	5	Moderate Low	5	Moderate Low	5
7	Upper Lower	4	Poor	4	Poor	4
8	Lower	3	Low	3	Low	3
9	Miserable Lower	2	Normal	2	Normal	2
10	Most Lower	1	Very Low	1	Very Low	1

Table 4
Value Matrix [24]

Experience (E)	Value	Influence (I)	Value	Interest (IN)	Value
10 Years	10	Absolutely High	10	Absolutely High	10
9 Years	9	Very High	9	Very High	9
8 Years	8	High	8	High	8
7 Years	7	Moderate High	7	Moderate High	7
6 Years	6	Medium	6	Medium	6
5 Years	5	Moderate Low	5	Moderate Low	5
4 Years	4	Poor	4	Poor	4
3 Years	3	Low	3	Low	3
2 Years	2	Normal	2	Normal	2
1 Years	1	Very Low	1	Very Low	1

Sub-step 2.3: Input values to all parameters for each of stakeholder.

In this sub-step, PO enters the values for all parameters for each of the stakeholder involved in the project. The values are based on the stakeholder prioritisation model and the value matrix. Table 5 shows an example of entering values for all parameters for each stakeholder based on the stakeholder prioritisation model.

Table 5
Stakeholder prioritisation model with values

SLNo	S Name	D	E	IR	P	DK	TS	I	IN
1	S1	Scrum Master (10)	10Y (10)	Absolutely High (10)	Absolutely High (10)	Absolutely High (10)	Absolutely High (10)	Very High (9)	Absolutely High (10)
2	S2	CEO (10)	12Y (10)	Absolutely High (10)	Absolutely High (10)	Absolutely High (10)	Absolutely High (10)	Absolutely High (10)	Absolutely High (10)
3	S3	Assistant Manager (7)	7Y (7)	High (8)	High (8)	High (8)	High (8)	High (8)	Very High (9)
4	S4	Manager (9)	6Y (6)	Absolutely High (10)	Absolutely High (10)	High (8)	High (8)	Very High (9)	Absolutely High (10)

Sub-step 2.4: Compute the stakeholder's weights using fuzzy inference rule.

In sub-step 2.4, the parameters were compared using 'fuzzy logic' to see how they related to each other. Each group was then subjected to a union and intersection analysis. When only parameters with the highest values are taken into account, the best outcome from the related types of parameters is found using the union operation, and the best outcome from all parameters is found using the intersection operation. Table 6 shows the fuzzy operations that apply the fuzzy inference rule to determine the links between each criterion for each stakeholder involved.

Table 6
Fuzzy operation [25]

No	Group Wise Operation	Output	Final Output (Finishing the intersection operation)	Comment
1	Influence ∪ Power	Preferred value is the highest (max)	Common value of all parameters will be preferred (min)	The designation can take into consideration to determine the priority of those stakeholders if more than one value is comparable.
2	Interest ∪ Technology Skill	Maximum value is preferable (max)		
3	Interpersonal Relationship ∪ Experience	Highest value is preferred (max)		
4	Domain Knowledge ∩ Designation	The preferred value is common (min)		

Table 7 illustrates a fuzzy operation with designation and domain knowledge for each of the stakeholder involved in a project.

Table 7
An illustration of fuzzy operation with designation and domain knowledge

SLNo	D = Designation	DK= Domain Knowledge	A = D ∩ DK
1	10	10	10
2	10	9	9
3	7	8	7
4	9	8	8

Table 8 illustrates a fuzzy operation with power and influence for each of the stakeholder involved in a project.

Table 8
An illustration of fuzzy operation with power and influence

SLNo	P = Power	I= Influence	B = P ∪ I
1	10	9	10
2	10	10	10
3	8	8	8
4	10	9	10

Table 9 illustrates a fuzzy operation with interest and technology skill for each of the stakeholder involved in a project.

Table 9
An illustration of fuzzy operation with interest and technology skill

SLNo	IN = Interest	TS= Technology Skill	C = IN ∪ TS
1	10	9	10
2	10	10	10
3	9	8	9
4	10	9	10

Table 10 illustrates a fuzzy operation with interpersonal relationship and experience for each of the stakeholder involved in a project.

Table 10
An illustration of fuzzy operation with interpersonal relationship and experience

SLNo	IR = Interpersonal Relationship	E= Experience	D = IR ∪ E
1	10	10	10
2	10	10	10
3	9	7	9
4	10	6	10

Table 11 illustrates the priority settings for all stakeholders involved in a project and their final weights and ranks for all stakeholders based on the fuzzy operations. The results of all operations (from Table 6 to Table 9) are included in the final table (Table 10) to obtain the results of the intersection operation (minimum values) that provides the best values of the stakeholders involved.

Table 11
An illustration of setting prioritisation for all stakeholders

SLNo	A	B	C	D	R = A ∩ B ∩ C ∩ D
1	10	10	10	10	10/10 = 1.0% (Rank 1)
2	9	10	10	10	9/10 = 0.9% (Rank 2)
3	7	8	9	9	7/10 = 0.7% (Rank 4)
4	8	10	10	10	8/10 = 0.8% (Rank 3)

Based on the final relative weights or values in Table 11, the prioritised stakeholders are listed in Table 12 according to their ranking.

Table 12
Example of setting prioritisation for all stakeholders

SLNo	SName
1	S1
2	S2
3	S4
4	S3

Sub-step 2.5: Give each stakeholder its appropriate amount of weight.

The selected stakeholders can be ranked in this sub-step according to the values of these stakeholders. To prioritise the user stories, a relative weight is assigned to each stakeholder based on the data obtained in this sub-step. When many stakeholders represent the same value, the designation or position within a certain organisation is taken into account to determine which stakeholder takes precedence.

Step 3: PO identifies all functional user stories (FUs) and non- functional user stories (NFUs).

Finding the FUs and NFUs that should be incorporated into a software system with the highest priority is the third step of the i-USPA process. Here, the numbers n and m represent the number of FU and NFU candidates, respectively. Suppose there are n FU candidates, such as FU1, FU2, FU3,…. FUn, and m NFU candidates, such as NFU1, NFU2, NFU3,..., NFUm, which must be ranked using i-USPA. The general rule is that there should be 5 to 15 requirements or user stories per sprint, with 20 requirements or user stories being the maximum. For the case of ATM system, 10 FUs and 5 NFUs are thus identified as requirements to be prioritised. The example of FUs is shown in Table 13.

Table 13
Example of the FUs of the ATM

FU ID	FU Description
FU1	As a Customer I want to Login to my account using card and PIN code So that I can perform the transactions
FU2	As a Customer I want to check the balance of my bank account So that I can perform transactions.
FU3	As a Customer I want to withdraw cash from my bank account through ATM So that I may save my time.
FU4	As a Customer I want to deposit cash in my bank account through ATM So that I may save my time and perform transactions later.
FU5	As a Customer I want to deposit cheque in my bank account through ATM So that I may save my time and perform transactions later.
FU6	As a Customer, I want to transfer money from my account to another bank account through ATM, so that I may save my time.
FU7	As a Customer, I want to view transaction history for my bank account through ATM, so that I can check the history of the transaction.
FU8	As a Customer, I want to print transaction receipt for my bank account through ATM, so that I can refer to it later.
FU9	As a Customer, I want to change PIN number for my bank account through ATM, so that I can maintain the security of my ATM account
FU10	As a Customer, I want to logout from my bank account through ATM, so that I may end up my ATM session.

Only two categories of patterns for non-functional requirements (NFR), which are security design patterns and fault tolerance design patterns are considered in this study. The fault tolerance design patterns contribute to several software quality attributes such as reliability, availability, usability and performance. These quality attributes are the critical factors that should be considered at an early stage of software development. The reasons why these two NFR patterns were selected in this study are because most of the functions in a software system or software product must be aligned with these two NFRs to produce a high-quality software product and to ensure that the released software product meets the needs of the users [26]. The NFR patterns used in this study have been adopted from the research of [26]. The selected patterns were also tested with ten Scrum team members during the research done by [27]. Examples of NFUs are listed in Table 14.

Table 14
Example of the NFUs of the ATM based on NFR patterns

NFU ID	Name	Description
NFU1	Availability	The percentage of time that the software system is in operation to provide its intended function
NFU2	Security	The extent to which access to the desired function by unauthorized persons ca n be controlled while still providing its function to users
NFU3	Usability	The extent to which a user is able to understand, learn, use and being attracted to a function
NFU4	Performance	The extent to which how fast the system can interact with the user to perform the desired function
NFU5	Reliability	The extent to which the system can be expected to perform its intended funct ion with required precision

Step 4: Apply Planning Game (PG) to all FUs & divide them into 3 sections.

A powerful multi-criteria decision-making technique, Saaty's AHP, has been used in several situations across a wide range of fields [28,29,33]. AHP is recognised as the best method since it produces reliable results, straightforward to use, error-tolerant, and it is based on a ratio scale [33]. However, AHP suffers from scalability and time consumption [29]. As a result, the PG technique has been used in this study to divide all FUs into three divisions based on their business value, which helps to minimise complexity, solve the scalability problem in AHP, and save time. The software planning process in the context of ASD mostly uses the PG technique [30].

Figure 3 shows the Step 4 of the proposed i-USPA. In this step, PO selects the FUs and divides them into three sections according to the business value. Table 15 shows an example of dividing ten FUs into three sections according to their importance. For the first section, all selected FUs are prioritised and estimated in the first sprint. The remaining FUs are prioritised and estimated in the following sprints.

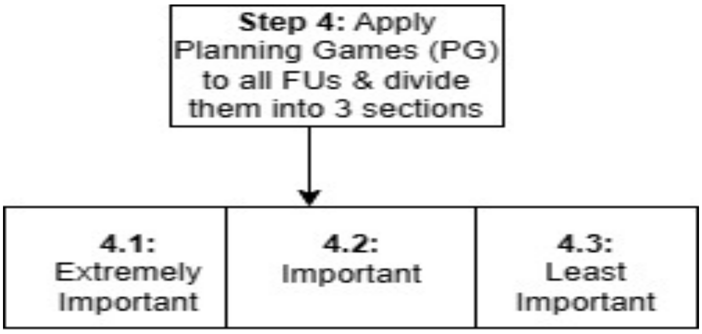

Fig. 3. Sub-steps of step 4

Table 15
Example of the NFUs of the ATM based on NFR patterns

Section	Section 1: Extremely Important (a need and business goal)	Section 2: Important	Section 3: Least Important
No of User Story (Selected by PO)	5	3	2

Step 5: Produce Pairwise Comparison Matrices.

A decision matrix is produced by setting the matching FUs in rows and the NFUs in columns. The *n*FUs and *m*NFUs are inserted into the decision matrix's rows and columns, to produce the $n \times m$ decision matrix D. D has been built as described in Table 16 using the previous step's example as a guide. For the first sprint, only five FUs are prioritised according to the FUs selected by PO for the first section (extremely important, a need, and a business objective).

Table 16
Example of decision matrix, *D*

	NFU1	NFU2	NFU3	NFU4	NFU5
FU1	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}
FU2	D_{21}	D_{22}	D_{23}	D_{24}	D_{25}
FU3	D_{31}	D_{32}	D_{33}	D_{34}	D_{35}
FU4	D_{41}	D_{42}	D_{43}	D_{44}	D_{45}
FU5	D_{51}	D_{52}	D_{53}	D_{54}	D_{55}

Step 6: Select a pair of FUs & NFUs, determine and calculate the degree of importance of an NFU for a given FU.

In this step, the decision-makers will rate the importance of each NFU for a given FU. FUs are impacted by NFUs. As a result, the main objective of this phase is to determine the value of the impact. NFUs can have an impact on FUs in different ways. One of them is they can affect the priority or value of a user story. For instance, if a NFU is that the system must offer 99.999% uptime, then user stories that improve reliability or fault tolerance may be more important than user stories that add new features. The nominal scale and the actual scale are two scales used by the i-USPA to evaluate this extent. The nominal scale is an interface scale that is used to simplify the i-USPA for communication with decision-makers so that they are not required to understand the details of the actual scale. For internal calculations, however, i-USPA makes use of the actual numerical scale. Table 17 outlines the nominal and actual scale used by i-USPA.

Table 17
Nominal scale and actual scale of i-USPA

Nominal scale	Actual scale
Very High Importance (VHI)	1
High Importance (HI)	0.75
Low Importance (LI)	0.5
Very Low Importance (VLI)	0.25
No Importance (NI)	0.001

Each decision maker is given a pair of FU and NFU (chosen from the rows and columns of matrix D, respectively), and is then given a series of tasks with the goal of determining the importance level of each NFU for attaining each associated FU. A nominal scale is used to assign each pair (FU and NFU), indicating the degree of significance between the linked user stories. To complete the components of the decision matrices as established in step 5, decision matrix D1, D2, D3, and D4 represent the opinions of five stakeholders. Table 18 provides a sample of these values. Note that the figures enclosed in brackets correspond to the actual scale values.

Table 18
Example decision matrix with nominal scale values

Decision Matrix	FUs	FUs				
		NFU1	NFU2	NFU3	NFU4	NFU5
D1	FU1	VHI (1)	VHI (1)	VHI (1)	VHI (1)	VHI (1)
(Stakeholder 1)	FU2	HI (0.75)	VHI (1)	HI (0.75)	HI (0.75)	VHI (1)
	FU3	HI (0.75)	VHI (1)	VHI (1)	VHI (1)	VHI (1)
	FU4	HI (0.75)	VHI (1)	VHI (1)	VHI (1)	VHI (1)
	FU5	LI (0.5)	VHI (1)	HI (0.75)	HI (0.75)	VHI (1)
D2	FU1	HI (0.75)	LI (0.5)	LI (0.5)	HI (0.75)	VHI (1)
(Stakeholder 2)	FU2	LI (0.5)	LI (0.5)	HI (0.75)	LI (0.5)	HI (0.75)
	FU3	HI (0.75)	HI (0.75)	HI (0.75)	VHI (1)	VHI (1)
	FU4	LI (0.5)	VHI (1)	HI (0.75)	HI (0.75)	VHI (1)
	FU5	VHI (1)	LI (0.5)	VHI (1)	LI (0.5)	LI (0.5)
D3	FU1	HI (0.75)	VHI (1)	HI (0.75)	VHI (1)	VHI (1)
(Stakeholder 3)	FU2	HI (0.75)	VHI (1)	HI (0.75)	HI (0.75)	HI (0.75)
	FU3	VHI (1)	VHI (1)	HI (0.75)	VHI (1)	HI (0.75)
	FU4	VHI (1)	VHI (1)	HI (0.75)	VHI (1)	VHI (1)
	FU5	HI (0.75)	VHI (1)	VHI (1)	LI (0.5)	HI (0.75)
D4	FU1	VHI (1)	VHI (1)	LI (0.5)	HI (0.75)	VHI (1)
(Stakeholder 4)	FU2	HI (0.75)	HI (0.75)	LI (0.5)	LI (0.5)	HI (0.75)
	FU3	VHI (1)	VHI (1)	HI (0.75)	HI (0.75)	VHI (1)
	FU4	VHI (1)	VHI (1)	LI (0.5)	HI (0.75)	VHI (1)
	FU5	HI (0.75)	VHI (1)	HI (0.75)	LI (0.5)	HI (0.75)

Step 7: Determine the ranking of NFUs in relation to all FUs using the TFN and alpha cut technique.

Once all pairings have been examined, i-USPA employs the TFN and alpha cut technique to prioritise NFUs in relation to all FUs, reflecting the weighting for Step 8. The i-USPA generates a prioritised list of NFUs by assessing the overall degree of relevance of each NFU with reference to all FUs. An NFU may be classified as a high priority NFU if it achieves the highest overall relevance score among all FUs. The subsequent sub-steps demonstrate how to calculate the NFUs' priority vector in detail. The sub-steps for Step 7 of the proposed i-USPA are shown in Figure 4.

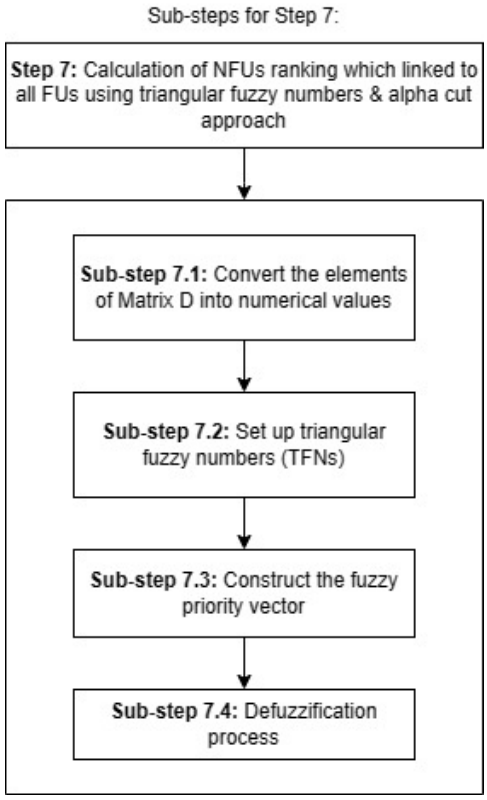

Fig. 4. Step 7 sub-steps

Sub-step 7.1: The elements of matrix D are converted into numerical values.

The decision matrix D is created by first converting all values given with the nominal scale of i-USPA into the actual corresponding scales.

Sub-step 7.2: The triangular fuzzy numbers (TFNs) should be setup.

The TFN is calculated to combine the various relevance levels of each NFU for various FUs. The fuzzy set theory can be used by TFN to merge the concepts of a decision-maker. TFN has been used in this investigation because it is the type of fuzzy number used most frequently [32]. The following equations serve as representations for the TFN, Tx_i:

$$T_{xi} = \left(L_{xi,}, M_{xi}, H_{xi}\right), i = 1..m, \text{and } L_{xi}, M_{xi}, H_{xi} \in [0.001,1] \quad (1)$$

$$Mx_i = \sqrt[n]{D_{x_{i}a}.D_{x_{i}b}.D_{x_{i}c} \ldots D_{x_{i}n}} \quad (2)$$

where Lx_i and Hx_i represent the lowest and highest value of non-functional user story *"x_i"*, respectively; Mx_i will be generated by calculating the geometric mean of all values belonging to the non-functional user story *"x_i"*; m is the total number of NFUs; *n* is the total number of FUs; and $Dx_i\ a$ specifies an opinion of a decision maker toward the importance degree of the non-functional user story *"x_i"* for achieving the functional user story *"a"*.

Sub-step 7.3: The fuzzy priority vector's construction.

After calculating the TFN value for each NFU, the fuzzy priority vector, namely $\tilde{F}x$, is generated as illustrated in Table 19. The values of ($\tilde{F}x$) are derived from Eq. (1):

Table 19
Fuzzy priority vector, $\tilde{F}x$

x_1	x_2	....	x_m
T_{x_1}	T_{x_2}	....	T_{x_m}

Sub-step 7.4: Process of defuzzification.

The defuzzification process is carried out by i-USPA using the alpha cut technique [33]. Using Eq. (3), the estimated TFN values are transformed into quantifiable values, and the defuzzification is completed to provide the priority vector W:

$$W(\tilde{F}x_i) = [\beta \times f_\alpha(L_{x_i}) + (1-\beta) \times f_a(H_{x_i})], 0 \leq \alpha, \beta \leq 1 \quad (3)$$

where *fα* (Lx_i) = (Mx_i - Lx_i) ×α+ Lx_iwhich indicates the left-end border value of alpha cut for; Fx_i; and *fα* (Hx_i) = Hx_i -(Hx_i-Mx_i) ×α, which shows the right-end border value of alpha cut for. Finally, using the following Eq. (4), the vector NW of normalised weights is produced by normalising the estimated priority vector, W:

$$NW_j = \frac{W_j}{\sum_{j=1}^{m} W_j} \quad (4)$$

A decision maker is given a prioritised list of NFUs together with their key values in relation to all current FUs by using the aforementioned procedures. The TFN and an alpha cut technique are utilised in this stage to rank the NFUs that are connected to all FUs using the Integrated User Story Prioritisation Approach supporting tool (i-USPT). Fuzzy numbers are used to lessen the different kinds of ambiguity and competing requirements during a bid. This is done in order to produce reliable work. The rule is to produce an inference output based on the input variables. The "degree of truth", as opposed to "true or false" standard, is employed in this method of counting. The values of the priority vector of NFUs with regard to all FUs are illustrated in Table 20, in which the values of the matrix *D's* elements are created and converted to actual scales.

Table 20
An illustration of how to compute NFUs priority vectors in relation to all FUs

Decision Matrix	NFUs	Fx	W	NW
D1	NFU1	(0.25, 0.60415, 1.0)	0.61	0.145
	NFU2	(0.75, 0.981004, 1.0)	0.93	0.221
	NFU3	(0.75,0.794417,1.0)	0.83	0.198
	NFU4	(0.75,0.794417,1.0)	0.83	0.198
	NFU5	(1.0,1.0,1.0)	1.0	0.238
D2	NFU1	(0.25,0.511916,1.0)	0.568	0.178
	NFU2	(0.25,0.521829,1.0)	0.573	0.180
	NFU3	(0.25,0.60415,1.0)	0.615	0.193
	NFU4	(0.25,0.650059,1.0)	0.64	0.201
	NFU5	(0.5,0.828354,1.0)	0.79	0.248
D3	NFU1	(0.5, 0.699457, 1.0)	0.72	0.227
	NFU2	(0.5, 0.819025, 1.0)	0.78	0.246
	NFU3	(0.001,0.095002,1.0)	0.3	0.095
	NFU4	(0.5,0.675479,1.0)	0.71	0.224
	NFU5	(0.25,0.704971,1.0)	0.66	0.208
D4	NFU1	(0.5,0.803466,1.0)	0.777	0.222
	NFU2	(0.5,0.844394,1.0)	0.80	0.229
	NFU3	(0.25,0.608913,0.75)	0.55	0.157
	NFU4	(0.5,0.557091,0.75)	0.59	0.169
	NFU5	(0.5,0.812619,1.0)	0.781	0.223

Step 8: Calculate the FUs Ranking.

Other stakeholders may participate in the prioritization process. The final prioritized lists of FUs and NFUs may differ depending on the perspectives of many stakeholders involved. To merge the prioritized lists of FUs provided by different stakeholders into a single list, a weighted average matrix must be created. Using Eq. (5), it is demonstrated how the priority vector R is obtained from the weights in Sub-Step 7.4.

$$R_i = \prod_{j=1}^{m} \mathcal{D}_{ij}^{NW_j}, i = 1..n \quad (5)$$

The resulting vector R is then normalized to create the normalized priority vector of functional requirements, NR, which ensures that the ranking values in the end will fall between 0 and 1:

$$NR_i = \frac{r_i}{\sum r_i} \quad (6)$$

The ultimate ranking is shown by the decreasingly ordered FUs, with the FU with the highest NR value being the most crucial. Table 21 shows an example of calculation of FUs priority vectors with respect to NFUs. These examples are from five stakeholders involved in a project.

Table 21
Example of FUs priority vectors with respect to NFUs are calculated

Stakeholder	FUs	NFUs						
		NFU1	NFU2	NFU3	NFU4	NFU5	R (Eq 5)	NR (Eq 6)
		0.145	*0.221*	*0.198*	*0.198*	*0.238*		
S1	FU1	1	1	1	1	1	1.000	0.079
	FU2	0.75	1	0.75	0.75	1	0.865	0.067
	FU3	0.75	1	1	1	1	0.964	0.075
	FU4	0.75	1	1	1	1	0.964	0.075
	FU5	0.5	1	0.75	0.75	1	0.807	0.063
		0.178	*0.180*	*0.193*	*0.201*	*0.248*		
S2	FU1	0.75	0.5	0.5	0.75	1	0.692	0.072
	FU2	0.5	0.5	0.75	0.5	0.75	0.598	0.062
	FU3	0.75	0.75	0.75	1	1	0.853	0.089
	FU4	0.5	1	0.75	0.75	1	0.789	0.082
	FU5	1	0.5	1	0.5	0.5	0.505	0.052
		0.227	*0.246*	*0.095*	*0.224*	*0.208*		
S3	FU1	0.75	1	0.75	1	1	0.912	0.095
	FU2	0.75	1	0.75	0.75	0.75	0.805	0.084
	FU3	1	1	0.75	1	0.75	0.917	0.096
	FU4	1	1	0.75	1	1	0.973	0.102
	FU5	0.75	1	1	0.5	0.75	0.755	0.079
		0.222	*0.229*	*0.157*	*0.169*	*0.223*		
S4	FU1	1	1	0.5	0.75	1	0.854	0.077
	FU2	0.75	0.75	0.5	0.5	0.75	0.657	0.059
	FU3	1	1	0.75	0.75	1	0.910	0.082
	FU4	1	1	0.5	0.75	1	0.854	0.077
	FU5	0.75	1	0.75	0.5	0.75	0.748	0.067

Step 9**:** To obtain at final rankings, combine prioritised lists of FUS & NFUs provided by a number of stakeholders.

The final prioritised lists of FUs and NFUs take into account the various points of view of the many stakeholders. The Weighted Average (WA) technique will be used to incorporate the prioritised lists of FUs and NFUs that have been provided by various stakeholders into the suggested methodology. The Weighted Average (WA) technique [31] was used to aggregate prioritised lists of FUs and NFUs that were provided by various stakeholders; Eq. (7) is used to determine the final weight for each FU.

$$UR_i = \sum_{j=1}^{k} NR_{ij} * W_{s_j}, i = 1..n \quad (7)$$

Table 22 shows the example of calculating the prioritized list of FUs. The final weights and rank are based on the stakeholders' weights from the fuzzy operations in Step 2 of i-USPA.

Table 22
Example of calculating the prioritized list of FUs

	Stakeholders' weights (from Fuzzy Operations)	*1.0*	*0.9*	*0.7*	*0.8*	Final Weights & Rank
FUs		S1	S2	S3	S4	UR
FU1		0.079	0.072	0.095	0.077	0.272(3)
FU2		0.067	0.062	0.084	0.059	0.229(4)
FU3		0.075	0.089	0.096	0.082	0.288(1)
FU4		0.075	0.082	0.102	0.077	0.282(2)
FU5		0.063	0.052	0.079	0.067	0.219(5)

The single prioritized list of NFUs must be created using the same procedure. Eq. (8) has been applied to determine a final weight for each NFU associated with the prioritizing issue:

$$UW_i = \sum_{j=1}^{k} NW_{ij} * W_{s_j}, i = 1..m \tag{8}$$

Table 23 shows an example of the calculation of the prioritized list of NFUs. The final weights and rank are based on the weights of the stakeholders from the fuzzy operations in Step 2 of i-USPA.

Table 23
Example of calculating the prioritized list of NFUs

	Stakeholders' weights	*1.0*	*0.9*	*0.7*	*0.8*	Final Weights & Rank
NFUId	Name	S1	S2	S3	S4	UW
NFU1	Availability	0.145	0.178	0.227	0.222	0.188 (4)
NFU2	Security	0.221	0.180	0.246	0.229	0.218 (2)
NFU3	Usability	0.198	0.193	0.095	0.157	0.167 (5)
NFU4	Performance	0.198	0.201	0.224	0.169	0.196 (3)
NFU5	Reliability	0.238	0.248	0.208	0.223	0.231 (1)

Based on the prioritized list of NFUs, the development team will refer to this NFUs prioritized list and should give more attention to the NFU according to the rank given by all stakeholders involved in the project in order to produce a high-quality software and satisfy the user's needs.

Step 10**:** Estimate the user stories.

A story point method is employed in this study to estimate the user stories. The story points are calculated based on hours. A PO together with the development team will decide on story point for each of the FU (consider the NFU as well). Table 24 shows the example of user stories estimation using story points for the first sprint which involve only five FUs.

Table 24
Example of user stories estimation using story points

	Stakeholders' weights (from Fuzzy Operations)	*1.0*	*0.9*	*0.7*	*0.8*	Final Weights & Rank	Story Point
FUs		S1	S2	S3	S4	UR	
FU1		0.079	0.072	0.095	0.077	0.272(3)	8
FU2		0.067	0.062	0.084	0.059	0.229(4)	3
FU3		0.075	0.089	0.096	0.082	0.288(1)	3
FU4		0.075	0.082	0.102	0.077	0.282(2)	5
FU5		0.063	0.052	0.079	0.067	0.219(5)	5

4. Results and Discussion

4.1 Validate i-USPA through Expert Review

Five software practitioners or experts have reviewed the proposed approach. All have years of relevant experience and a variety of educational degrees, and all are subject-matter experts. Two of them are Scrum Masters, two are developers, and one is a PO. They have substantial experience in software development, with at least two to five years. The agile approach they most frequently utilise is Scrum, though one has also used DevOps.

The feasibility of merging functional and non-functional requirements in the Agile software development process was demonstrated by the findings of this study. The participants reported that integration has improved the overall quality of the software. Specifically, the integration of non-functional requirements helped to identify potential performance issues early in the development process, which reduced the likelihood of having to make costly changes later on. The participants also reported that the integration of functional and non-functional requirements facilitated ensuring that the software met both the functional requirements (e.g., the features that the software must provide) and the non-functional requirements (e.g., the performance, reliability, and usability of the software).

Besides that, the participants also agreed that by having stakeholder analysis during the prioritisation process also benefits the Agile software development process, since it helps in identifying the various stakeholders involved in the project and their respective demands, expectations, and concerns. Stakeholder analysis helps identify all the stakeholders who will be impacted by the software system. Stakeholder analysis helps to identify the needs and expectations of each stakeholder. These needs and expectations are used to prioritise user stories and requirements in the product backlog. Prioritising requirements based on stakeholder needs helps to ensure that the most important features are delivered first. In addition, the stakeholder analysis assists in balancing the needs of different stakeholders. This is important because different stakeholders may have conflicting needs or expectations. For example, end-users may want a system that is easy to use, while management may prioritise security and performance. By analysing stakeholder needs and expectations, scrum teams can find ways to balance these needs and satisfy all stakeholders.

During the study session, the suggested approach was initially explained to the software practitioners. They were then given broad questions concerning the advised course of action and asked to describe both its advantages and disadvantages. Their insightful feedback and recommendations were then noted. Table 25 shows the responses from the participants.

Table 25
The responses of the software experts or practitioners

Practitioner (P)	Quote
P1	*"It is a good initiative to focus on the prioritisation of both functional and non-functional user stories".*
P2	*"It will give a good impact on the practical side".* *"It can deliver successful software projects within the prescribed time and budget".*
P3	*"We need to know the non-functional requirements eventually; in some contexts, we need to know them very early in the project"* *"It could assist the project developer during their work too"*
P4	*"It is good that the stakeholders also can get involved in the prioritisation of the requirements to make sure that the requirements needed can be satisfied".* *"It will save the time and also the budget as well".*
P5	*"It is good to involve stakeholders during the prioritisation process to get their opinions".* *"Can increase the stakeholder's satisfaction".*

Only people with doctorates in computer science (CS), software engineering (SE), information technology (IT), or a closely related field who had worked on software projects utilising Agile methods for at least two to five years were regarded as experts or software practitioners in this study. As mentioned above, all five participants had experience with SE, notably with agile software development and process optimisation.

Additionally, all practitioners work for reputable software companies. The responses from the study's participants indicate that the prioritisation process is significantly impacted by the combination of both FUs and NFUs. Both sides benefit from the stakeholders' participation in the prioritisation process because it helps the developer better understand their needs while also saving time and money.

4.2 Validate i-USPA using Experimental Approach

This section presents the results of the user story prioritisation approach by validating the i-USPA using an experimental approach. There are three different approaches involved in the experiments: i-USPA, AHP-based, and conventional approaches. The study involves 5 experts and 38 final year students from the Faculty of Computer Science and Information Technology (FCSIT) at the Universiti Malaysia Sarawak (UNIMAS). The dependent variables considered are ease of use, usefulness, and level of agreement. The experiment compared the three approaches - i-USPA, AHP-based, and conventional approach in terms of their ease of use, usefulness, and level of agreement. Each participant was assigned a random approach to evaluate, ensuring a balanced comparison.

4.2.1 Ease of use

Both the experts and students reported that i-USPA was easier to use compared to the AHP-based and conventional approaches. The user-friendliness of i-USPA contributed to its positive reception among the participants, making it an appealing option for user stories prioritisation in the Scrum-Agile environment.

4.2.2 Usefulness

Participants found i-USPA was perceived as more useful compared to the AHP-based approach and the conventional approach. The main reason for this preference was i-USPA's provision of a stakeholder analysis as part of its approach. This additional analysis provided valuable insights for stakeholders, contributing to better decision-making during the user stories prioritisation process.

4.2.3 Level of agreement

The level of agreement was measured by comparing the prioritisation results of each approach with a goal standard. The i-USPA approach demonstrated a higher percentage of accuracy in the level of agreement compared to both the AHP-based approach and the conventional approach. This indicates that the user stories requirements sequence generated by i-USPA was closer to the goal standard, making it a more reliable, more consistent approach and producing accurate results, enhancing the reliability of the prioritisation outcomes.

Overall, the i-USPA demonstrated superiority in ease of use, usefulness, and level of agreement over the AHP-based and conventional approaches in the agile user stories prioritisation context. As the conclusion, the findings of this study suggest that i-USPA is a favorable option for user stories prioritisation in the agile environments. Its stakeholder analysis provision, and higher level of agreement contribute to its effectiveness in helping teams make informed decisions about which user stories to prioritise. As such, the positive feedback from both experts and final year students highlights the advantages of i-USPA as an efficient and effective approach for prioritising user stories in agile projects. The study's limited sample size of five experts and 38 final year students from a specific university (FCSIT, UNIMAS) may impact the generalizability of the findings. The study might not have accounted for potential biases or individual preferences of the participants towards certain approaches.

5. Conclusion and Future Work

This study is important because it benefits stakeholders and promotes knowledge in many different fields. The agile Scrum development methodology, frequently employed in software projects, is the primary subject of this study, which primarily focuses on how Scrum impacts the timely and cost-effective completion of software projects. This study contributes to the body of knowledge on SE by analysing the influence of the Agile Scrum technique on the prioritisation of NFUs, a subject which has been neglected in the SE community and enables efficient delivery and higher-quality software projects. In Agile software development, user stories and requirements are prioritised based on their business value and stakeholder needs. Stakeholder analysis helps identify the various stakeholders involved in the project and their demands and expectations. Nevertheless, stakeholder analysis is not a one-time activity, but is rather an ongoing process with continuous engagement with stakeholders throughout the project. Therefore, this study proposes that stakeholder analysis be included during the prioritisation process to fulfil their needs so that they will be satisfied with the final product. In conclusion, stakeholder analysis is an essential activity in user story or requirements prioritisation in agile software development. By understanding stakeholder needs and expectations, scrum teams can prioritise user stories or requirements based on business value and ensure that the most important features are delivered first. Additionally, stakeholder analysis helps both to build stakeholder trust and ensure continuous stakeholder engagement throughout the project.

For future study it should involve a larger and more diverse participant pool to validate the findings across different organizational settings and contexts. It is advisable to conduct similar experiments with professionals from various industries and levels of experience in agile methodologies to obtain a more comprehensive evaluation of the approaches. Other than that, further investigations into the factors affecting user stories prioritisation and their interactions with the chosen approaches could provide deeper insights into the decision-making process.

Acknowledgement
The authors would like to thank Universiti Putra Malaysia (UPM) (Fundamental Research Grant Scheme (FRGS), FRGS/1/2020/ICT01/UPM/02/2) and Universiti Malaysia Sarawak (UNIMAS), for supporting this research.

References
[1] Razali, Rozilawati, and Fares Anwar. "Selecting the right stakeholders for requirements elicitation: a systematic approach." *Journal of Theoretical and Applied Information Technology* 33, no. 2 (2011): 250-257.
[2] Preiss, Otto, and Alain Wegmann. "Stakeholder discovery and classification based on systems science principles." In *Proceedings Second Asia-Pacific Conference on Quality Software*, pp. 194-198. IEEE, 2001.
[3] Lim, Soo Ling, Daniela Damian, and Anthony Finkelstein. "StakeSource2. 0: using social networks of stakeholders to identify and prioritise requirements." In *Proceedings of the 33rd international conference on Software engineering*, pp. 1022-1024. 2011. https://doi.org/10.1145/1985793.1985983
[4] Olaronke, Iroju, Ikono Rhoda, and Gambo Ishaya. "An appraisal of software requirement prioritization techniques." *Asian Journal of Research in Computer Science* 1, no. 1 (2018): 1-16. https://doi.org/10.9734/ajrcos/2018/v1i124717
[5] Ahmad, Sabrina, Riftika Rizawanti, Terry Woodings, and Intan Ermahani A. Jalil. "MCBRank Method to Improve Software Requirements Prioritization." *International Journal of Advanced Computer Science and Applications* 13, no. 7 (2022). https://doi.org/10.14569/IJACSA.2022.0130728
[6] Garg, Nikita, Pankaj Agarwal, and Shadab Khan. "Recent advancements in requirement elicitation and prioritization techniques." In *2015 International Conference on Advances in Computer Engineering and Applications*, pp. 237-240. IEEE, 2015. https://doi.org/10.1109/ICACEA.2015.7164702
[7] Krishnan, M. Soumya. "Rfp based requirement prioritization–a one-step solution." *Materials Today: Proceedings* 5, no. 1 (2018): 642-649. https://doi.org/10.1016/j.matpr.2017.11.128
[8] Bajaj, Punam, and Vineet Arora. "Multi-person decision-making for requirements prioritization using fuzzy AHP." *ACM SIGSOFT Software Engineering Notes* 38, no. 5 (2013): 1-6. https://doi.org/10.1145/2507288.2507302
[9] Al-Ta'ani, Rami Hasan, and Rozilawati Razali. "A framework for requirements prioritisation process in an agile software development environment: empirical study." *International Journal on Advanced Science, Engineering and Information Technology* 6, no. 6 (2016): 846-856. https://doi.org/10.18517/ijaseit.6.6.1375
[10] Alrashoud, Mubarak, Etessam Hazza, Fayez Alqahtani, Muneer Al-Hammadi, Abdolreza Abhari, and Ahmed Ghoneim. "Cognitive and hierarchical fuzzy inference system for generating next release planning in saas applications." *IEEE Access* 7 (2019): 102966-102974. https://doi.org/10.1109/ACCESS.2019.2929214
[11] Masood, Maria, Farooque Azam, Muhammad Waseem Anwar, and Anam Amjad. "Defining meta-model for value-oriented requirement prioritization technique." In *Proceedings of the 7th International Conference on Computer and Communications Management*, pp. 72-77. 2019. https://doi.org/10.1145/3348445.3352739
[12] Chopra, Raj Kumar, Varun Gupta, and Durg Singh Chauhan. "Experimentation on accuracy of non functional requirement prioritization approaches for different complexity projects." *Perspectives in science* 8 (2016): 79-82. https://doi.org/10.1016/j.pisc.2016.04.001
[13] Rahman, Marfizah Abdul, Rozilawati Razali, and Dalbir Singh. "A Risk Model of Requirements Change Impact Analysis." *J. Softw.* 9, no. 1 (2014): 76-81. https://doi.org/10.4304/jsw.9.1.76-81
[14] Noviyanto, Fiftin, Rozilawati Razali, and Mohd Zakree Ahmad Nazree. "Understanding requirements dependency in requirements prioritization: a systematic literature review." *International Journal of Advances in Intelligent Informatics* 9, no. 2 (2023): 249-272. https://doi.org/10.26555/ijain.v9i2.1082
[15] Pa, Noraini Che, and Abdullah Mohd Zain. "A Survey of Communication Content in Software Requirements Elicitation involving Customer and Developer." *J. Softw. Syst. Dev* 2011 (2011): 1-12. https://doi.org/10.5171/2011.742200
[16] Gupta, Ankita, and Chetna Gupta. "Towards dependency based collaborative method for requirement prioritization." In *2018 Eleventh International Conference on Contemporary Computing (IC3)*, pp. 1-3. IEEE, 2018.

https://doi.org/10.1109/IC3.2018.8530542

[17] Sheemar, Heera, and Gurpreet Kour. "Enhancing user-stories prioritization process in agile environment." In *2017 International Conference on Innovations in Control, Communication and Information Systems (ICICCI)*, pp. 1-6. IEEE, 2017. https://doi.org/10.1109/ICICCIS.2017.8660760

[18] Ismail, Fatin Filzahti, Rozilawati Razali, and Zulkefli Mansor. "Considerations for cost estimation of software testing outsourcing projects." *Int. J. Adv. Sci. Eng. Inf. Technol* 9, no. 1 (2019): 142-152. https://doi.org/10.18517/ijaseit.9.1.6382

[19] McZara, Jason, Shahryar Sarkani, Thomas Holzer, and Timothy Eveleigh. "Software requirements prioritization and selection using linguistic tools and constraint solvers—a controlled experiment." *Empirical Software Engineering* 20 (2015): 1721-1761. https://doi.org/10.1007/s10664-014-9334-8

[20] Freeman, R. Edward. *Strategic management: A stokcholder approach*. Pitman, 1984.

[21] Hujainah, Fadhl, Rohani Binti Abu Bakar, Basheer Al-Haimi, and Mansoor Abdullateef Abdulgabber. "Investigation of stakeholder analysis in requirement prioritization techniques." *Advanced Science Letters* 24, no. 10 (2018): 7227-7231. https://doi.org/10.1166/asl.2018.12919

[22] Ramzan, Muhammad, M. Arfan Jaffar, and Arshad Ali Shahid. "Value based intelligent requirement prioritization (VIRP): expert driven fuzzy logic based prioritization technique." *International Journal Of Innovative Computing, Information And Control* 7, no. 3 (2011): 1017-1038.

[23] Voola, Persis, and A. Vinaya Babu. "Requirements uncertainty prioritization approach: a novel approach for requirements prioritization." *Softw Eng Int J (SEIJ)* 2, no. 2 (2012): 37-49.

[24] Babar, Muhammad Imran, Masitah Ghazali, Dayang NA Jawawi, Siti Maryam Shamsuddin, and Noraini Ibrahim. "PHandler: an expert system for a scalable software requirements prioritization process." *Knowledge-Based Systems* 84 (2015): 179-202. https://doi.org/10.1016/j.knosys.2015.04.010

[25] Pacheco, Carla, and Ivan Garcia. "Effectiveness of stakeholder identification methods in requirements elicitation: experimental results derived from a methodical review." In *2009 Eighth IEEE/ACIS International Conference on Computer and Information Science*, pp. 939-942. IEEE, 2009. https://doi.org/10.1109/ICIS.2009.100

[26] Majumdar, Shariful Islam, Md Saidur Rahman, and Md Mijanur Rahman. "Stakeholder prioritization in requirement engineering process: a case study on school management system." *Computer Science and Engineering* 4, no. 1 (2014): 17-27.

[27] Amorndettawin, Methinee, and Twittie Senivongse. "Non-functional requirement patterns for agile software development." In *Proceedings of the 2019 3rd International Conference on Software and e-Business*, pp. 66-74. 2019. https://doi.org/10.1145/3374549.3374561

[28] Achimugu, Philip, Ali Selamat, Roliana Ibrahim, and Mohd Naz'ri Mahrin. "A systematic literature review of software requirements prioritization research." *Information and software technology* 56, no. 6 (2014): 568-585. https://doi.org/10.1016/j.infsof.2014.02.001

[29] Dabbagh, Mohammad, Sai Peck Lee, and Reza Meimandi Parizi. "Functional and non-functional requirements prioritization: empirical evaluation of IPA, AHP-based, and HAM-based approaches." *Soft computing* 20 (2016): 4497-4520. https://doi.org/10.1007/s00500-015-1760-z

[30] Misaghian, Negin, and Homayun Motameni. "An approach for requirements prioritization based on tensor decomposition." *Requirements Engineering* 23, no. 2 (2018): 169-188. https://doi.org/10.1007/s00766-016-0262-6

[31] Bukhsh, Faiza Allah, Zaharah Allah Bukhsh, and Maya Daneva. "A systematic literature review on requirement prioritization techniques and their empirical evaluation." *Computer Standards & Interfaces* 69 (2020): 103389. https://doi.org/10.1016/j.csi.2019.103389

[32] Mahmood, Yasser A., Alireza Ahmadi, Ajit Kumar Verma, Ajit Srividya, and Uday Kumar. "Fuzzy fault tree analysis: a review of concept and application." *International Journal of System Assurance Engineering and Management* 4 (2013): 19-32. https://doi.org/10.1007/s13198-013-0145-x

[33] Liou, Tian-Shy, and Mao-Jiun J. Wang. "Ranking fuzzy numbers with integral value." *Fuzzy sets and systems* 50, no. 3 (1992): 247-255. https://doi.org/10.1016/0165-0114(92)90223-Q

[34] Govil, Nikhil, and Ashish Sharma. "Estimation of cost and development effort in Scrum-based software projects considering dimensional success factors." *Advances in Engineering Software* 172 (2022): 103209. https://doi.org/10.1016/j.advengsoft.2022.103209

[35] Chow, Tsun, and Dac-Buu Cao. "A survey study of critical success factors in agile software projects." *Journal of systems and software* 81, no. 6 (2008): 961-971. https://doi.org/10.1016/j.jss.2007.08.020