INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND EMERGING TECHNOLOGIES ISSN: 2522-3348
(IJCET)- VOL4(2)- pg.7-14
DECEMBER 2020

https://ijcet.salu.edu.pk Publisher: Shah Abdul Latif University Khairpur

Prioritization Of Software Requirements Through Hierarchical Aggregation

Atif Ali[1]¸ Muhammad Imran Naz[2*], Yasir Kamal[3], Syed Salman Ahmed[4], Saima Farooq[5]
[1]PMAS Arid Agriculture University, Rawalpindi, Pakistan
[2,3,4]Shah Abdul Latif University Khairpur, Sindh, Pakistan
[5]Military College of Signals NUST, Islamabad, Pakistan

Abstract

A fundamental goal of any engineering and software, including, in particular, is to achieve quality products. To meet these needs is essential to conduct formal requirements engineering and especially their prioritization. Prioritization of software requirements is a complex decision-making process. Existing models lack adequate flexibility and adaptability to the explicit contexts of organizations. In this work, a method for prioritizing requirements that uses *aggregation operators is hierarchically* proposed. The system provided for the consideration of topics such as requirements significance and relativity. A case study shows the proposal's relevance. The article ends with suggestions for potential research that will help to improve the method's validity.

Keywords: prioritization requirements, aggregation operators, engineering requirements, Software requirement gathering

I INTRODUCTION

In particular software engineering, the ultimate goal of any engineering is the quality of the final product. The software and information system's quality is often determined by the ability to satisfy quality attributes based on the information on customers and end users' needs, as obtained and software requirements specified or derived from them. A correct definition and requirements analysis is one factor contributing to software projects' success [1].

The method of prioritizing requirements (PR) is a difficult one. It is determined that functionalities are appropriate to include in each release of the software product to be developed. *"The challenge is to select the 'right' requirements out of a given superset of candidate requirements so that all the different key interests, technical constraints, and preferences of the critical stakeholders are fulfilled, and the overall business value of the product is maximized."* Many authors have treated it as one activity with higher complexity levels in requirements engineering and fundamental to projects' success [2].

Major Aspects of Prioritization: Requirements can be prioritized, taking many different aspects into account. An aspect is a property or attribute of a project and its requirements that can be used to prioritize requirements. Often, aspects interact, and fluctuations in one aspect could impact another aspect's Importance. The stakeholders should prioritize which requirements are of utmost importance for the system. The importance is multifaceted and could be the urgency of implementation, importance for product architecture, strategic importance [1,2].

A. ***Penalty***: It is possible to evaluate the penalty that is introduced if a requirement is not fulfilled. The penalty is not just the opposite of importance.
B. ***Cost***: The developing organization usually estimates the implementation cost. Measures that influence cost include the complexity of the requirement, the ability to reuse existing code, the amount of testing and documentation needed. Cost is often expressed in terms of staff hours.
C. ***Time***: It is influenced by many other factors such as degree of correspondence in development, training needs, need to develop support infrastructure, complete industry standards.
D. ***Risk:*** Every project carries some amount of risk.
E. ***Volatility:*** Volatility of requirements is considered a risk factor and is sometimes handled as part of the risk aspect. A point of view is to handle them separately [3].
F. ***Other Aspects***: Financial benefit, strategic benefit, competitors, competence/resources, release theme, ability to sell. Stakeholders need to develop a list of aspects to help in the decision-making process.

*Corresponding Author: *atif.alii@yahoo.com

G. ***Combining Different Aspects***: The purpose of any prioritization is to assign values to distinct prioritization objects that allow the establishment of a relative order between the set's objects. In this case, the objects are the requirements that need to prioritize. The prioritization can be done with various measurement scales and types [2,3].

Prioritization Models: The minimum robust prioritization scale is the ordinal scale, where the requirements are well-ordered so that it is likely to see which requirements are more critical than others, but not how much more critical. A commonly used model for PR includes the following steps that may be iterated throughout the life cycle [4]:

- Selection of one or more prioritization criteria,
- Assigning values to the criteria selected by one or more involved,
- Aggregation criteria to achieve a final order requirement.

In the initial stages of a software development project, requirements are generally inaccurate. Insofar as the project progresses and understanding grows of the product requirements specified in greater detail. Prioritization of requirements is a process that can be done at different times of the lifecycle, with requirements at different levels of abstraction. The requirements are prioritized, taking into account different variables imposed by organizations' needs and context. Researchers define a set of variables, including acute adverse effects, cost, time, risk, and volatility. Wieger proposes a method based on value, cost, and risk requirements. Authors classify variables in business aspects (e.g., Competition of the market, regulations), customer satisfaction, and technical aspects (e.g., Cost of development). These criteria are added in existing methods to calculate a unique priority value assigned to each requirement. However, aggregation criteria are not sufficiently flexible and adaptable to specific contexts of organizations. Aggregation and fusion of information from different sources to provide a single output that somehow synthesizes the original information and supports the decision is the active work area [5].

This study aims to present a method for prioritizing requirements based on aggregation operators' use hierarchically for merging information.

II. LITERATURE REVIEW

The work focused on the analysis of specific characteristics that show the notorious differences enter methods. Following each method are briefly described and then analyzed characteristics detail.

A. *Analytical Hierarchy Process (AHP):*

This method focuses on evaluating an attribute to decide, for example, importance, benefits, criticality, etc. The requirements are located in a matrix of NxN (N: number of requirements). The customer compares all requirements, assigning a value to each other on a scale of 1 to 10. Upon completing the matrix, the values are added for each row and divided by the number of requirements. The result is the estimated value of the attribute for each requirement. The more requirements, the more comparisons you can get an order of requirements to implement if it is based on relative importance [6].

In the Wiegers units prioritizing, which may be requirements, use cases, etc., which depends on the project's scale, are listed. They are estimated for each unit: Profit, Penalty, implementation cost, and risk, using a scale from 1 to 9. Each attribute may be assigned a numerical weight. It is the calculated priority of each unit weighting the four attributes and their respective weights, and the order of implementation of the units to prioritize is obtained.

$, Hs, etc., or just 100 points distributed among all the requirements: In the method, 100-Point Test, an absolute numerical initial scale of 100 units (coded number), such as is used. This is done by several different clients or developers, depending on the attribute to prioritize. The larger units assigned requirements are the most important, and ranking weapon or can be grouped. We must be aware that customers should have a global vision of the company and its real problems, not distribute them according to their own needs [2,6].

B. *Numerical Assignment Method:*

This is based on establishing a qualitative scale such as desirable or optional critical priority for defining attributes. Existing requirements between qualitative groups are distributed. Certain restrictions may help, for example, that each group cannot be less than a certain number of requirements. Each requirement within a group has the same priority.

C. *Ranking Method:* It is based on a client or developer or a group of them with the same rank within the organization choose an attribute on which prioritizing requirements and then assign each requirement an ordinal numeric priority 1 to N (N: number of requirements). In the Top-Ten Requirements method, each customer or developer puts together a list of ten requirements that they believe higher priority over an attribute to define. Given the variation between each list created, a balance is made, and a consolidated list of the most important requirements regarding the attribute is created [7].

D. *The Method Based on Objectives and Scenarios:* It consists of breaking down the system subgoals' main objectives. Each sub-objective about urgent implementation and subsequently priority is assigned to each sub-object satisfies the scenario. Priority is transferred from the sub-scenarios objectives and requirements of these scenarios extracted. A list of prioritized requirements used to determine the order to be implemented is obtained.

E. The Software Quality Function Deployment (SQFD): This method tries to define the requirements to invest resources during development. It includes the following tasks to prioritize: identify types of customers; assign a weight of importance to each type of client; define the needs of the business and organize them into a hierarchy of needs considering their interdependencies; prioritizing the needs of the business for each type of client based on their interests as meet and weigh the weight of the type of customer; needs to align with the requirements [8].

F. Easy Win-Win: This method, based on negotiation, establishes a compromise between customers and developers where it is proposed that all commit and that "everyone wins" (hence, the term win-win). IT tools are used to support this approach in collecting, processing, prioritizing, and negotiating "win conditions" (interests involved), having as attributes the importance and difficulty of implementation. It's about getting what win conditions should be taken into account. The following tasks are included: Stakeholders express their topics of interest (services, interfaces, non-functional aspects, restrictions on the project, and development issues, among others). Through brainstorming, each involved anonymously gives opinions and ideas on the topics. A list of "win conditions" based on the ideas and opinions grouped by topics is armed consensus. Each involved a numerical scale calls the "win conditions" considering two attributes, importance to the business (given by customers) and difficulty of implementation (given by developers). Based on the values assigned to both attributes by all involved, the "win conditions" you are grouped into [9]:

- Very important and easy to implement,
- Very important and challenging to implement,
- Not important and easy,
- Not important and difficult. Involved analyzing the "win conditions" in the resulting groups, mainly considering those with low consensus to negotiate disagreements. For "win conditions," conflicting options are proposed to solve them,

G. Win-Win Quantitative Method: It is a variation of the above. Prioritization steps:

Identify the complete set of requirements and requirements subsets containing it.

- By AHP, a weight is assigned to each class involved in prioritizing.
- By AHP, the importance of each subset of requirements is computed from the point of view of each class involved. Here, N vectors of each subset's importance (number of classes involved N) are obtained.
- Is obtained a vector of the importance of subsets requirements, taking into account each subset's relative importance for each class involved and each class's weight.
- The importance of each requirement is obtained from inner subassemblies, multiplying the subset's importance to which the requirement belongs the relative importance of the requirement within the class.
- For each subset of requirements, is calculated by developers the effort required to develop them. These six steps are performed iteratively and incrementally. The requirements engineer determines the number of iterations to achieve an optimal level of refinement [9,10].

Requirements Triage: The concept of "Triage" involves deciding what requirements will be implemented and which not needed to be fully satisfied, important, and desirable to be satisfied are separated. Important are those who need treatment to see what to do and what to leave for the subsequent iterations. This and importance by customers through negotiation are based on the developer's implementation effort and implementing each requirement iteration. Using a probabilistic graph shows the probability of success according to the decisions taken regarding the analysis and defined times.

H. Value-Oriented Prioritization (VOP) Method: The core values of the company (example: To be leaders in this aspect in the market, high Sales, Marketing, etc.) are defined in, and customers negotiate, they assign them a numerical value based on the business value of 0 (not important) to 10 (critical). The risks are identified and weighed with a negative value. Each requirement is analyzed for the principal value of every risk identified, putting a weight. A weighted formula is used to calculate the absolute priority of each requirement. Finally, a ranking of requirements is armed with the order of implementing them [11].

I. MOSCO: The customer’s requirements separate into four groups according to their importance: Must (critics) Should (important) Could (desirable), and it would (achievable). It negotiates with customers through interviews to determine what requirements to implement and whatnot.

J. Cost-Value method: Using AHP to calculate the customer's benefit and cost by developers of each requirement concerning others. Each requirement is positioned on a graph of cost (x-axis) and value (y-axis), displayed which requirements have less cost index and high index value. These requirements will be implemented. With the graph, you can put together a ranking.

K. Pirogov Method: It automates much of prioritization, manages dependencies between requirements through shared characteristics, and allows trading between stakeholders to set various criteria for prioritizing. The requirements are placed by automatic algorithms clustered by several characteristics, iteratively merging clusters until it reaches an appropriate granularity level. The algorithm places each requirement in a single cluster, but the same may naturally belong to more than one, then the degree of proximity of each requirement for each cluster is calculated. Clusters are prioritized by a collaborative technique (as Easy Win-Win), where the differences involved negotiating level clusters. Each requirement is prioritized [12].

Unlike other comparison methods work prioritization of requirements, it has been considered a diverse number of characteristics that distinguish each method. It is believed that they allow a comparison on how to use the methods and for what purpose. Below are listed the nine characteristics analyzed.

It is considered a distinctive feature to take into account or not the viewpoint of customers and developers. Depending on which attributes are graded on the method, this will determine whether only customers or developers are also prioritizing. Attributes, such as "benefit of implementing the requirement," tend to be rated by customers, while for an attribute as "development effort requirement," developers often intervene. Also, there may be a group of clients participating where each assigns an independent value to the same attribute ("Other" in Table 1), or where negotiate among themselves to obtain a single value for the attribute ("One negotiated" in table 1), or you may have a single point of view given by a single person ("a" in table 1) [11,13].

Another feature is analyzed to prioritize the unit where it has been observed that it is generally the individual requirements. However, some methods prioritize more complex elements such as use cases, user stories, business interests, etc.

Attributes and Diff Methods:

The attributes considered in the different methods are as varied. Some methods use a single attribute while others combine several, for example, ***Urgency implementation***, Benefit of implementation, cost of implementation, technical risk, importance, or just some methods leave freedom of the requirements engineer choosing the appropriate attributes for prioritizing.

Generally, the purpose of prioritizing is to determine the order to implement the standards previously defined. To know at what stage of the life cycle of the software development is to implement a requirement or set of related requirements, which has to do with prioritization but at the same time with some negotiation between clients and developers. It is to reach the best possible solution where the customer is satisfied, considering the time and resources allocated to the project. Still, many methods have their order of priority, which are very different, determining the relative importance of the requirement or selecting the optimal set of requirements to be implemented in the next iteration [14].

It has been taken into account in this study if the method considers the interdependence of requirements. It is understood that the methods that take into account arrive at much more accurate and actual results than those who do not consider since it is often not feasible to implement a requirement without a previously implemented, or that the joint implementation of two requirements you can reduce costs and facilitate the implementation itself.

It is significant to consider the different types of scale castoff for attributes and similar priorities. We can name in general use numerical scale and qualitative scale. The latter refers to organize the requirements in different groups, such as those Priority High, Medium, or Low. Regarding the numerical scale, we can subdivide it into three major groups [15]:

a. Absolute numerical, involving specific, quantifiable units, such as hours of work, money, etc.
b. Numerical proportional dealing a mathematical calculation to arrive for example, at a ratio or a percentage and, they are usually used various weights and/or more attributes;
c. Ordinal number, based on the sequentially ordered requirements. Some methods use various types of scales,

Finally, it has taken into account the complexity of the algorithm prioritization, ranging from low to high complexity with multiple nuances. This complexity is given by the number of calculations and/or comparisons and/or attributes and weights required by the method.

AGGREGATION OPERATIONS

The method of combining modified data to generate a single output is known as knowledge aggregation. Aggregation operators are a type of mathematical function that can be used to combine data. In a domain D, N values are combined, and a value in that domain is returned..

Denoting these functions $\mathbb{C}$, the aggregation operators are functions of the form [16]:

$$C: Nn \rightarrow (1)$$

Aggregation operators are useful in several cases. It is critical in decision-making for assessing and designing alternatives. Each of the operator families has its own set of characteristics that can be used to model different scenarios. You may use the weighted average (WA) to give information sources a weight representing their reliability or importance/preference. Meanwhile, the operator family OWA (ordered weighted averaging, or average weighted ordinate in Spanish) compensates for or gives weight to data based on their values. Modeling redundancy, complementarity, and interactions between parameters is possible with diffuse integrals. Operates, on the other hand, are not appropriate for expressing the properties specified in human reasoning. [16,17].

2.1 Half Power

The average aggregation operator heavy power (weighted mean power, WPM) can express the degree of simultaneity and relative rank of entries (weights). Furthermore, it allows the structure of hierarchical aggregation models to be developed. The following is the description of the rth WPM:

$$M_n^{[r]}(a,w) = \left(\sum_{i=1}^{n} a_i^r w_i\right)^{1/r}$$

Where $W_i \in [0,1]\ Y \sum_{i=1}^{n} w_i = 1Yr$

It can be nominated to reach desired properties logically. Defining weights corresponding to each feature and sub-features using AHP is possible. The WPM average is calculated using the aggregation logic model ranking of preferences, a hierarchical model (LSP for its acronym in English). One of the LSP model's major strengths is its ability to model various logical relationships b/w attributes and sub-features to represent different participants' needs in the evaluation process. In table 1, the main operators are shown [18].

Types of Requirement		Level of Intensity	Symbol	Orness	Andness	Exponent
Disjunctive Requirement		Strongest	D	1.000	0	+∞
		Very Strong	D++	0.9375	0.0625	20.63
		Strong	D+	0.8750	0.1250	9.521
		Medium Strong	D+-	0.8251	0.1875	5.802
		Medium	DA	0.7510	0.25	3.92
		Medium Weak	D-+	0.69	0.31	2.79
(Partial Disjunctive)		Weak	D-	0.61	0.37	2.01
		Very Weak	D--	0.55	0.43	1.44
Neutrality			A	0.510	0.413	1.414
Disjunctive Requirement	Non Mandatory	Very Weak	C--	0.43	0.56	0.61
		Weak	C-	0.37	0.62	0.26
	Mandatory Requirement	Medium Weak	C-+	0.31	0.68	-0.14
		Medium	CA	0.251	0.751	-0.721
		Medium Strong	C+-	0.181	0.811	-1.651
(Partial Disjunctive)		Strong	C+	0.121	0.871	-3.151
		Very Strong	C++	0.06	0.93	-9.06
		Strongest	C	0	1.0	-∞

Table 1. Values of functions Conjunction / Disjunction Generalized

In the aggregation phase, the decider may use two parameters:

- Simultaneity (the degree to which two things happen at the same time (andness).
- The entry's relative value (weights).

These operators have the exciting feature of allowing you to add details (D, D-, C+, A…) etc denoted as symbols to weigh the intensity level as long as you know which elements are required and optional. In the author's view, all of the previously discussed elements could represent a more practical prioritization of requirements.

III. PROPOSED METHOD

The following tasks are included in the proposed method: selection criteria and requirements, obtaining information, standardization of values, weights, and aggregation determine vectors. Below is presented graphically (Figure 1) the activities in the workflow and each described [2,18]:

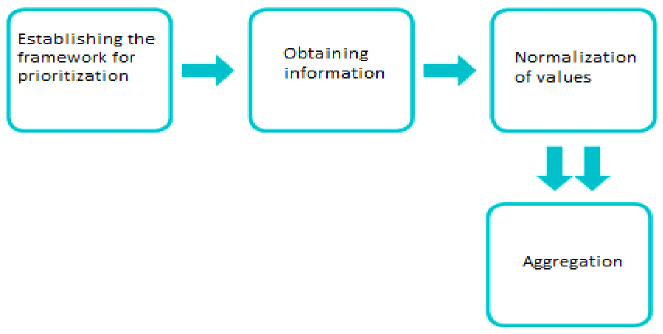

Fig. 1. Activities of the workflow for prioritizing requirements

1. Establishing the framework for prioritization: are selected criteria and requirements to be evaluated. Where C = {C1,C2,C3,….Ck, ...,} with K ≥ 2} with the criteria to be evaluated, and R = { r1,r2,r3,….rj } With j ≥ 2 requirements.

2. Collecting information: Collected information on decision-makers' interests. This data reflects the evaluation of each criterion in terms of the parameters. The utility vector is written as Vj= Vj1,Vj2,Vj3,....Vjj, where Vjk is the preference for

the criterion Ck" of the requirement. Rj. The assessment is given in the interval [0,1], 0 being the worst value and the best one [19].

3. Normalization of values: Values are normalized preferences considering whether they are of benefit type or cost. . Being + ' "the standard value is calculated for this type benefit criteria such as:

$$v_{jk} = \frac{v_{k\min} - v_{jk}}{v_{k\min} - v_{j\max}} \qquad (3)$$

and for the cost types:

$$v_{jk} = \frac{v_{k\max} - v_{jk}}{v_{k\max} - v_{j\min}} \qquad (4)$$

Vk min is the minimum valuation criteria regarding Vk max: It is the maximum assessment regarding criterion k.

4. Aggregation: The aggregation function; OAG:[0 ,1] n --> [0,1] is got by a process of hierarchical aggregation. Used WPM (1) using the logic model rating preference (LSP by its acronym) because this is adjusted in a more realistic vocational guidance process. Aggregation operators hierarchically gives elasticity to the method. The possibility of obtaining the decision maker's preferences and expression vectors weight directly is another of its strengths [20].

IV. CASE STUDY

Then a case study is presented with the fundamental purpose of showing the pertinency of the proposal. Developing the criteria for selected technical difficulty, cost, and value for the project of a set of five software requirements about an information system. Subsequently, the evaluation is performed for each requirement concerning the selected criteria (Table 2) [21].

Requirement	Technical Difficulty	Cost	Value
R1	0.9	0.2	0.7
R2	0.2	0.4	0.7
R3	0.3	0.6	0.8
R4	0.4	0.8	0.3
R5	0.6	0.7	0.7

Table 2. Assessment requirements

Criteria related to the technical difficulty and cost are cost criteria type and normalized according to (4). The criterion value is normalized to benefit type (3). Normalization results are shown in Table 3.

Requirement	Technical Difficulty	Cost	Value
R1	0.00	1.00	0.80
R2	1.00	0.67	0.80
R3	0.86	0.33	1.00
R4	0.71	0.00	0.00
R5	0.43	0.17	0.80

Table 3. Standardization of criteria

The structure of hierarchical aggregation is obtained. Aggregation operators reflecting simultaneity as established LSP [27, 28] were used.

Initial Tickets	Operator		ID Block	Operator		ID Block
Cost	0.3	C- -	Cost/ Benefit	0.8	C-	Global Priority
Value	0.7					
Technical Difficulty				0.2		

Table 4. Aggregation Structure

The aggregation criteria's results allow for sorting requirements. The priority order, in this case, is as follows: R2> R3> R5> R1> R4>.

Requirement	AG
R1	0.365
R2	0.804
R3	0.788
R4	0.002
R5	0.543

Table 5. Results of aggregation

Specialists benefit from the technique's relative simplicity and the high versatility provided by the aggregation model used. Due to advances in decision-making software engineering, the results also demonstrate the applicability of presenting models to aid decision-making based on the topic's aggregation of knowledge, timeliness, and relevance. [2,21].

V. RESULTS AND DISCUSSION

Above are some important enumerations of literature work in Hierarchical Aggregation requirements prioritization. The work done in the past does not focus much on the

Hierarchical Aggregation of requirements; still, there is a lot of effort to be done in this regard. For requirement prioritization, various factors are considered: time, penalty, volatility, cost, benefit, a combination of other aspects, etc. Further study is required to look into numerous factors and interdependence among different like stakeholders, choosing the system's core requirements, Planning, and selecting plan, the optimal set of software requirements for implementation in consecutive releases. There is a need for work on reprioritization of requirements concerning the Analytical Hierarchy Process (AHP).

Future work has to focus on a combination of Top-Ten Requirement, Scale, Sophistication Scalability, Top-Ten Requirements, 100-Dollar Test. In future work may involve the fusion of Hierarchical Aggregation with artificial intelligence in a particular neural network to gather the benefits of both the approaches (Hierarchical Aggregation and Neural Networks)

VI. CONCLUSIONS

A framework for prioritizing requirements based on aggregation operators for combining data is discussed in this paper. WPM was used to build a hierarchical aggregation operator. Selection criteria, finding knowledge about decision-makers' preferences, standardization of values, and ultimately aggregating the normalized values of preferences are all part of the process. The ability to model the criterion and recompense value is the method's key benefit. Act with the fuzzy linguistic approach and multi-expert approach as future work is outlined. Another task is to create a software tool that aids in the work environment.

REFERENCES

[1]. Waheed, S., Hamid, B., Jhanjhi, N. Z., Humayun, M., & Malik, N. A. (2019). Improving knowledge sharing in distributed software development. IJACSA) International Journal of Advanced Computer Science and Applications, 10(6).

[2]. Ali, A., Hafeez, Y., Hussain, S., & Yang, S. (2020). Role of Requirement Prioritization Technique to Improve the Quality of Highly-Configurable Systems. *IEEE Access*, *8*, 27549-27573.

[3]. Ayub, K., Azam, F., Anwar, M. W., Amjad, A., & Jahan, M. S. (2019). A novel approach for software requirement prioritization based upon non functional requirements. *2019 7th International Conference in Software Engineering Research and Innovation (CONISOFT)*. https://doi.org/10.1109/conisoft.2019.00013

[4]. Ejaz, K., & Amjad, A. (2018). Model and technique over software requirement prioritization. *Pakistan Journal of Engineering, Technology & Science*, *6*(2). https://doi.org/10.22555/pjets.v6i2.1962

[5]. Hamid, M. A., Hafeez, Y., Hamid, B., Humayun, M., & Jhanjhi, N. Z. (2020). Towards an effective approach for architectural knowledge management considering global software development. International Journal of Grid and Utility Computing, 11(6), 780-791.

[6]. Sandanasamy, A., & Selvi, R. T. (2018). Software requirement prioritization a critical study on its techniques. *International Journal of Computer Sciences and Engineering*, *6*(11), 203-206. https://doi.org/10.26438/ijcse/v6i11.203206

[7]. Ali, A., Hafeez, Y., Abbas, S. F., & Sarwar, A. (2018, March). REQUIREMENTS PRIORITIZATION: A COMPARISON BETWEEN TRADITIONAL AND AGILE (SCRUM AND FDD). In *16th International Conference on Statistical Sciences* (p. 77).

[8]. Gupta, V., Chauhan, D., & Dutta, K. (2013). Regression testing based requirement prioritization of desktop software applications approach. *International Journal of Software Engineering and Its Applications*, *7*(6), 9-18. https://doi.org/10.14257/ijseia.2013.7.6.02

[9]. Kronberger, T., & Papakonstantinidis, L. (2019). “The win-win-win Papakonstantinidis model”: Bargaining possibilities when there are three involved parties on a labour market and two of them are active decision-makers – Cases Greece-Germany. *INTERNATIONAL JOURNAL OF INNOVATION AND* ECONOMIC DEVELOPMENT, 4(6), 68-98. https://doi.org/10.18775/ijied.1849-7551-7020.2015.46.2005

[10]. Krishnamoorthi, R., & Sahaaya Arul Mary, S. (2009). Factor oriented requirement coverage based system test case prioritization of new and regression test cases. Information and Software Technology, 51(4), 799-808. https://doi.org/10.1016/j.infsof.2008.08.007

[11]. Myo, W. W., Wettayaprasit, W., & Aiyarak, P. (2018). A noble feature selection method for human activity recognition using linearly dependent concept (LDC). Proceedings of the 2018 7th International Conference on Software and Computer Applications. https://doi.org/10.1145/3185089.3185131

[12]. Sandanasamy, A., & Selvi, R. T. (2018). Software requirement prioritization a critical study on its techniques. International Journal of Computer Sciences and Engineering, 6(11), 203-206. https://doi.org/10.26438/ijcse/v6i11.203206

[13]. YE, R., JIN, Z., WANG, P., ZHENG, L., & YANG, X. (2010). Approach for autonomous web service aggregation driven by requirement. Journal of Software, 21(6), 1181-1195. https://doi.org/10.3724/sp.j.1001.2010.03666

[14]. Jang, H., & Lee, S. (2021). Restaurant customers segmentation based on serving robots" attributes evaluation. Korean Journal of Hospitality & Tourism, 30(1), 49-63. https://doi.org/10.24992/kjht.2021.1.30.01.49

[15]. Sanayei, P. A., & Saneian, Z. S. (2013). Analysis of traditional attributes and website attributes in order to improve customers trust in electronic banking (The case of customers of Mellat bank, Iran, Shiraz branch). International Journal of Academic Research in Business and Social Sciences, 3(11). https://doi.org/10.6007/ijarbss/v3-i11/321

[16]. Exponential operations and an aggregation method for single-valued Neutrosophic numbers in decision making. (2017). *Information*, *8*(2), 62. https://doi.org/10.3390/info8020062

[17]. Tu, Z., & Xu, X. (2016). Requirement pattern elicitation approach of massive customers in domain oriented service requirement engineering. Enterprise Interoperability VII, 53-75. https://doi.org/10.1007/978-3-319-30957-6_5

[18]. Lu, Z., & Ye, J. (2017). Single-valued Neutrosophic hybrid arithmetic and geometric aggregation operators and their decision-making method. *Information*, *8*(3), 84. https://doi.org/10.3390/info8030084

[19]. Xu, J., & Pottinger, R. (2014). Integrating domain heterogeneous data sources using decomposition aggregation queries. *Information Systems*, *39*, 80-107. https://doi.org/10.1016/j.is.2013.06.003

[20]. Yu, X., & Xu, Z. (2013). Prioritized intuitionistic fuzzy aggregation operators. *Information Fusion*, *14*(1), 108-116. https://doi.org/10.1016/j.inffus.2012.01.011

[21]. Garg, U., & Singhal, A. (2017). Software requirement prioritization based on non-functional requirements. *2017 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence*. https://doi.org/10.1109/confluence.2017.7943258