ReQueCLASS: A FRAMEWORK FOR CLASSIFYING REQUIREMENT ELICITATION QUESTIONS BASED ON KIPLING'S TECHNIQUE AND ZACHMAN'S ENTERPRISE FRAMEWORK - A GUIDE FOR SOFTWARE REQUIREMENT ENGINEERS

By

SALIHU ABDULKADIR * HAMZAT OLANREWAJU ALIYU **

* Department of Computer Science, Federal University of Technology, Minna, Nigeria.
** School of Information and Communication Technology, Federal University of Technology, Minna, Nigeria.

Date Received: 11/01/2019 Date Revised: 02/02/2019 Date Accepted: 05/03/2019

ABSTRACT

At present, interview is still considered one of the pragmatic approaches to gathering software requirements from the different stakeholders in a software project. Despite unrelenting efforts by researchers, requirements gathered using this method still suffer anomalies such as inconsistency and incompleteness; this problem is partly due to communication gaps between Requirement Engineers (RE) and project stakeholders and partly due to the RE directing some questions to the wrong persons. This paper proposes a framework, which mirrors the Zachman's Enterprise Framework to systematically classify requirement interview questions and assign different question categories to appropriate persons in a disciplined way. A working software project is used as an example to illustrate the use of the framework.

Keywords: ReQueClass, Requirement Question Classification, Requirements Elicitation, Kipling Method, Zachman's Enterprise Framework.

INTRODUCTION

Software requirements elicitation is the process of finding out the requirements for an intended software system by communicating with the appropriate stakeholders in the project (Mohanan & Samuel, 2016; Sommerville, 2005). It is the basic and most important aspect in the requirement engineering process; doing it inaccurately will have many adverse effects on the intended software like questionable software quality, late delivery, and software that cost more than its worth (Wong, Mauricio, & Rodriguez, 2017; Jabar, Ahmadi, Shafazand, Ghani, & Sidi, 2013; Zhaoyin, Yanfang, & Chao, 2009).

Gathering information from different stakeholders is challenging (Sutcliffe & Sawyer, 2013) and this leads to a more complex and subjective task; inconsistency and incompleteness in requirement specification are constantly encountered in defining and interpreting of stakeholders' needs by REs.

Furthermore, stakeholder needs change at a great rate during software system development and both the stakeholders and REs rely solely on natural language communication (Mishra, Awal, & Elijah, 2017; Lee & Zhao, 2006). Therefore, it is of great importance that both parties use a clear form of communication to provide and analyze requirements that will lead to the precise definition of the problem.

Generally, it is a common knowledge that elicitation approaches, such as Interview and Questionnaires, depend on some well-defined questioning approaches to gather information. The literature acknowledges the fact that missing or wrongly-stated questions can lead to loss of information and consequently substandard requirement specifications (Nuseibeh & Easterbrook, 2000). Substandard requirement specifications are often characterized by inconsistency and/or incompleteness. Inconsistencies occur in the process of acquiring, specifying, and changing of goals from different stakeholders' requirements specifications (Mishra et al., 2017). The frequent changes in stakeholders' requirements have particular impact on the consistencies of the overall requirement specifications. In order to deal with the inconsistencies arising from such frequent changes, requirements specifications are at times, trimmed and this often leads to incompleteness.

From anecdotal evidences, incompleteness can also result from limited coverage of the questions being asked by the interviewees or the RE directing certain questions to the wrong persons and being unable to get useful responses; he may also get conflicting responses at times, thereby leading to inconsistency in the requirement so gathered.

A critical look at the aforementioned defects in requirement specifications will reveal to the reader, two fundamental causes of most of the problems – a) frequent changes in stakeholders' requirements and b) the RE not directing certain questions to the appropriate persons. At present, not much can be done about cause (a) as it has to do with human desire; we can, however, make efforts to address cause (b) by providing guides to the RE on the construction and assignment of questions. This paper proposes the Requirement Question Classification (ReQueClass) framework to guide the RE in the classification of software requirement interview questions into categories that are directed to some recommended persons. The authors are optimistic that by providing a guide to the RE on whom to direct which questions to during requirement elicitation, the framework will solve the aforementioned problems to a large extent.

The proposed framework takes cues from Kipling's questioning technique and the Zachman's Enterprise Framework; hence, the next section of this paper presents a short backgrounds on them (i.e., Kipling's technique and Zachman's framework) to help the reader follow the subsequent sections.

1. Background

1.1 Kipling's 5W1H Questioning Technique

The Kipling's 5W1H (What, Who, When, Where, Why, and How) technique was proposed in the early 1900s by Rudyard Kipling in a book entitled 'Just So Stories' (Kipling, 1902). His aim was to formulate a questioning technique that captures the majority needs of what people want to know about a news story.

The technique was later applied widely by journalists to report news (Carmagnola, 2008). Moreover, The Kipling's 5W1H method has been applied in medical field to diagnose and treat patients (Oranje, Al-Mutairi, & Shwayder, 2016), the approach helps physician to unmask information from patients completely. It has also been applied in academics to enhance students' composition writing ability such as narrative writing (Shabir, 2016) and to expand their ideas. Also important to this work is its application by Zachman (2003) to propose an enterprise architecture to explicitly document the information of an enterprise, which would normally have been kept in the heads of some certain individuals in the organization.

1.2 Zachman Enterprise Framework

The Zachman Enterprise Framework uses the Kipling's 5W1H method for classification of an organisation's architecture, which is represented graphically using a matrix, with columns and rows. It has been widely adopted by system analyst (Zachman, 2003) for information systems architecture. The framework helps an organisation to breakdown its processes as illustrated in Table 1.

2. Related Work

Different ways to perform requirement elicitation task have been described in the literature. However, due to the nature of the problems, one elicitation technique cannot work in all situations (Tiwari & Rathore, 2017). In this section, existing uses of the Kipling's 5W1H method in requirement elicitation are briefly reviewed.

Wong et al. (2017) have proposed a general framework to perform a systematic literature review on requirement engineering topics. They noted that there are no proposals about automation to support activities to define techniques, document, and refine requirements. Therefore, recommendations were made on proposals to cover the missing points in order to improve the requirements elicitation process.

Studies by (Mishra et al., 2017; Hanif et al., 2017; Jabar

	What (Data)	How (Function)	Where (Network)	Who (People)	When (Time)	Why (Motivation)
Scope	List of things important to the software project	List of processes the software performs	List of location in which the software operates	List of stakeholders important to the software	List of cycles significant to the software	List of software goals/strategies
Business Model	e.g. Semantic Model	e.g. Business process model	e.g. Business logistics Systems	e.g. Workflow model	e.g. Master schedule	e.g. Business Plan
System Model	e.g. Logical data Model	e.g. Applications architecture	e.g. Distributed Systems architecture	e.g. Human interface architecture	e.g. Processing structure	e.g. Business Rule Model
Technology Model	e.g. Physical data Model	e.g. System design	e.g. Technology Architecture	e.g. Presentation architecture	e.g. Control structure	e.g. Rule design
Detailed Representation	e.g. Data Definition	e.g. Program	e.g. Network architecture	e.g. Security architecture	e.g. timing definitions	e.g. Rule specification
Functioning System	e.g. Data	e.g. Function	e.g. Network	e.g. Organization	e.g. Schedule	e.g. Strategy

Table 1. Zachman Architecture Framework

et al., 2013) have proposed a refined means of requirement elicitation process.

However, the initial set of questions that maybe asked during requirement elicitation process for an intended software domain were not considered to capture stakeholders' views or concern.

Sadiq (2017) have presented a method to prioritize stakeholders' views on software requirements using a fuzzy-based approach. Stakeholders' prioritization plays an important role to detect which stakeholder's requirements is to be implemented in which phase of the software. This study, shows that in practical usage, same requirements are viewed in different ways by different stakeholders; linguistic terminologies may be used in describing the benefits of their requirements.

The Kipling's 5W1H approach, though, has been used severally for other reasons in software engineering (Chung, Won, Baeg, & Park, 2009). A study by (Lee & Zhao, 2006) apply the approach in re-documentation of a given legacy system with UML visual models. The six dimensions were mapped to domain-specific contexts as follows: 'Who' are the software developers (Developers role), 'Why' re-documentation (Benefits), 'What' are the use of UML elements in various stakeholders views (UML element use), 'Where' are the different views located (Use-case views of the legacy system), 'When' is the different phases going to take place (Time), 'How' to construct other elements views and relationships. It is undesirable to limit oneself to usage of only one form of requirement elicitation.

This paper proposes the Kipling's 5W1H approach (see Table 3) based on the Zachman enterprise application as another approach to construct questions that can be used to elicit information to improve the requirement elicitation process. To the best of the knowledge of the authors of this paper, there has not been existing work suggesting the application of the Kipling's 5W1H pattern and Zachman framework to provide a guide to REs on whom to direct which question to during requirement elicitation.

3. The ReQueClass Framework

The section presents the classification of Requirement Elicitation questions based on Kipling's 5W1H questioning technique, followed by the question sets in the later part of the section.

3.1 Kipling's 5W1H Classification

The Kipling's method utilizes the 5W1H questioning method. From the REs' perspective, to gather software requirement, the stakeholders should be provided with vital information on six dimensions (Oranje et al., 2016; Hart, 1996):

- Who (Actor)
- Why (Motivation)
- What (Data/Content)
- Where (Location)
- When (Time)
- How (Function)

Furthermore, an RE aims at gathering information for an intended software system to obtain a complete and consistent understanding of the problem domain by mapping out the classifications of work done within the scope of the software.

Recall from section 1 that the Kipling's 5W1H has been widely applied by journalists, medical practitioners, academics, and enterprise framework builders. Thus, journalists, medical practitioners, students, and REs share the goal of seeking to understand and report certain activities (respectively news events, diagnosis, and software requirements) completely.

Typically, Software requirement elicitation is all about exploring the needs of individual stakeholders for an intended system such as extracting and/or discovering needs of the users and other potential stakeholders and developing a concise requirement document (Hickey & Davis, 2003).

REs conduct requirement elicitations by following a well-formed protocol of elicitation processes (Pohl, 2016) to gather information and then analyze it to categorize findings based on a set of pre-proposed questions. Thus, the understanding of an intended software domain by a RE when performing a requirement elicitation is largely determined by the pre-proposed questions. Existing requirement elicitation guidelines (Nuseibeh & Easterbrook, 2000; Hadar, Soffer, & Kenzi, 2014; Sommerville, Sawyer, & Viller, 1998), suggest starting with an exploratory formulation of Questions by interacting with users for the intended software system. For REs knowledgeable in the domain, proposing a relevant set of coherent and probing questions thereafter may not be difficult; however, this task can be challenging to REs new to the domain.

Recall that the goal of the RE is to provide a well-defined rule for one to follow so that different REs can more or less produce similar results (so that the process can allow for knowledge re-use). Simply asking REs to explore some domain areas and developing an exploratory set of questions without a set of concrete guidelines may be too abstract and time consuming.

The Kipling's 5W1H technique provides six dimensions to completely report an area of interest. The authors of this paper argue that it can benefit REs by relieving their challenges in defining the initial set of questions. The next section elaborates on the authors' view on requirement elicitation question sets and how to apply the Kipling's 5W1H information gathering approach to classify them into categories.

3.2 Existing Question Set in the Literature

This subsection presents a survey of open questions asked by REs during requirement elicitation. Any software design starts with a defined design task, which is often specified by users in natural language. Therefore, the design task is user oriented and may be defined incorrectly or incompletely while the requirement elicitation process should generate formal and structured descriptions, which are engineering oriented. Asking the questions is an effective way to identify the user's real needs and to define a relatively more complete and consistent list of software requirement. Table 2 provides a set of major Open/Close ended question used during software requirement elicitation.

From Table 2, the authors derived some set of major questions used to capture stakeholders' views during requirement elicitation based on existing literatures. The existing works have proposed questions that can serve as an effective questioning set for requirements analysis to bridge the gap between problem and solution. The requirements engineer can use the question set pattern to analyze an intended software problem and assess any missing information. However, the set does not contain questions covering certain software aspects, such as goals of the system, functional limitations, stakeholders' level of expertise, deployment choice, version update, and how the system can change existing activities. These missing questions play important roles in providing a complete set of stakeholder requirements and effective mechanisms for development. This paper proposes an extension (see in Table 3) of the existing question set with suitable questions to address these concerns. Moreover, the existing question set (in Table 2) is not accompanied by a definite guide on who should be asked which question to get the most effective answer to guarantee the complete and consistent requirement specification. This concern is also addressed by the ReQueClass proposed in this paper.

	Questions from existing literatures	Authors
Q1	What triggered the need of the software? (The problems the solution will address)	(Reza, 2015; Wang & Zeng, 2009)
Q2	What is the domain of application of the intended software system?	(Reza, 2015)
Q3	What is the estimated population size of users for the software system?	(Reza, 2015; Abad, Shymka, Pant, Currie, & Ruhe, 2016)
Q4	What features do you need from the system?	(Abad et al., 2016; Wang & Zeng, 2009)
Q5	What are the existing features of the system if any?	(Abad et al., 2016)
Q6	What standards should the software conform to?	(Wang & Zeng, 2009)
Q7	Where will the system be deployed for usage?	(Reza, 2015; Wang & Zeng, 2009)
Q8	Where will the data produced by the system be saved?	(Chen & Jin, 2016)
Q9	Why do we need these functionalities?	(Abad et al., 2016; Chen & Jin, 2016)
Q10	Why does the problem occur?	(Wang & Zeng, 2009)
Q11	When will each stage of the project life cycle be completed?	(Abad et al., 2016; Wang & Zeng, 2009)
Q12	When will we be ready to start?	(Abad et al., 2016; Wang & Zeng, 2009)
Q13	When do you need the system to be ready?	(Reza, 2015; Wang & Zeng, 2009; Chen & Jin, 2016)
Q14	Who are the domain experts?	(Abad et al., 2016; Chen & Jin, 2016)
Q15	Who are the intended system users?	(Abad et al.2016; Chen & Jin, 2016
Q16	Who are the Res?	(Abad et al., 2016)
Q17	Who maintains the system?	(Abad et al., 2016)
Q18	Who among the users will the system improve?	(Wang & Zeng, 2009; Chen & Jin, 2016)
Q19	Who needs the software system to support his daily work?	(Chen & Jin, 2016)
Q20	Who are the main actors to use each of the functional requirement specified?	(Wang & Zeng, 2009)
Q21	How do you use the existing functionalities?	(Abad et al., 2016)

Table 2. Summary of Major Question Sets for Requirement Elicitation

3.3 Extended Question Set

Table 3 proposes an additional question set to complement the existing question set reported in Table 2; an effort towards improving on the completeness of software requirements.

A merger of the question sets in Tables 2 and 3, can be used to provide a complete set of information that are consistent. However, to consistently capture stakeholders' viewpoint or concerns requires grouping of questions based on who (Actors) to answer which question (Processes). To group the questions based on who to answer which question, Table 4 presents different stakeholders and questions to be assigned by each actor based on Zachman's enterprise framework. This approach will go a long way in resolving inconsistency in requirement specification by different users associated to the system.

The classification as depicted and grouped in Table 4 provides a descriptive view of different stakeholder viewpoints for requirement specification. Each cell in the table is unique and aligned with the cells immediately above and below it. Combination of the cells in one row provides a corresponding stakeholder.

3.3.1 Matrix Columns

The columns present the information that are asked of the software system based on the Kipling's 5W1H technique.

	Question Set	Importance
EQ1	What are your goals in developing this system?	A business objective that describe the future or desired result.
EQ2	What languages do you need for the software?	This is relevant when the organization has foreign nationals supporting them (Language Globalization).
EQ3	What are the different levels of stakeholders' expertise?	To describe the level of experience of each stakeholder in terms of the system.
EQ4	Why the deployment choice?	To understand why the system needs to be deployed in a par.
EQ5	How would you like to see the information provided by the system?	To give insight on how the output will be displayed.
EQ6	When will the software become obsolete?	This is the duration required for the software. For example, if the organization states they only need the software for a year, then a subscription model may be adequate.
EQ7	When will the software update occur?	To help software developers have insight into when the software version will be rolled out.

Table 3. Extended Question Set for Requirement Elicitation

These are:

- *What (Data/Content):* What is the software data, information or objects?
- *How (Functionalities):* How does the software work, i.e., what are the software's processes?
- *Where (Network):* Where are the software operations?
- *Who (Actors):* Who are the people that run the software, what are the software units and their hierarchy?
- *When (Time):* When are the software processes performed, i.e., what are the software schedules and workflows?
- *Why (Motivation):* Why are the processes, people, or locations important to the software?

3.3.2 Matrix Rows

Each row presents a unique view of the classification, from the views of a particular category of stakeholders. A row is assigned to each of the following stakeholders:

- *Domain Experts:* Understands the software domain scope and can offer a contextual view of the software.
- *Owner/Decision Maker:* Understands the business model and can provide a conceptual view of the software.
- *Requirement Engineer:* Analyze, design and develop the software system model and can build a physical and logical view of the software.
- *End User:* Provides a view of the functioning software system, from the perspective of a user (e.g., an end user, shareholder, or customer).

4. Discussion

4.1 Usage of Re-organizing Kipling's 5W1H Method based on Zachman Framework

The Kipling's 5W1H pattern (defined in Section 3 and shown in Table 2) can be used to formulate question sets to be asked by a stakeholder group. For example, users are asked questions to gather requirements, where as, the RE can be asked questions to analyze the requirements and bridge the gap between problem and intended solution. Generally, Requirements Elicitation is a widespread challenging task. The Kipling's 5W1H method offers a structured approach to tackle these challenges, as shown in Table 3. By answering these questions based on different users' viewpoints, the requirements engineer will be able to determine a complete and consistent set of requirement and reduce conflicts between different users' requirement.

4.2 Implementation

Based on section 3.1, the ReQueClass framework is applied to classify the requirement questions of a school management system software as presented in Table 5.

Conclusion

In Software Requirement Elicitation, a major problem is the process of providing an effective guide to requirement engineers to assign requirement elicitation questions to the right stakeholders in order to have a clear and complete requirement specification document. Essentially, this means that assigning the right question to the wrong user could lead to inconsistency and incompleteness in the software requirement specification document. This research proposes a framework which can serve as a guide to software RE on to whom to direct which question during Requirement Elicitation. Furthermore, the proposed framework uses the Zachman's framework for enterprise application and the Kipling's 5W1H information gathering approach to capture each stakeholders' viewpoint correctly and completely. The findings will be of great interest to REs who are new to a domain using the framework to serve as a guide to elicit information. Future work is to extend the technique to build a knowledge base

	What (Data/Content)	How (Functionalities)	Where (Network)	Who (Actors)	When (Time)	Why (Motivation)
Domain Expert	Q2, Q3, Q6, EQ1	_	Q7, Q8	Q15, Q16, Q17, Q19, Q20	EQ6, EQ7	Q10, EQ4
Owner/Decision Maker	Q1, EQ3	-	-	Q14, Q18	Q12, Q13	-
Requirement Engineer	-	-	-	-	Q11	-
End User	Q4, Q5, EQ2	Q21, EQ5	-	-	-	Q9

Table 4. Adaptation of The Zachman Enterprise Architecture Framework

	What (Data/Content)	How (Functionalities)	Where (Network)	Who (Actors/Stakeholders)	When (Time)	Why (Motivation)
Domain Expert	Q2: Educational Q3: 10,000 Above Q6: Agile Development EQ1: The goal is to have a robust system that support all device to successfully automate student record management system	EQ6: Several Years	Q7:Cross-Platform Support Q8: External Database	Q15: Students & ICT Staff Q17: JD Lab NG Q18: Students & ICT Unit Q19: ICT Staff Q20: Candidate App-Student Record Management System: IT Staff Courseware-Student. E-card-Student		Q10: Students information are been managed manually, so it takes lot of time toretrieve information. EQ4: To enable access on all device platform
Owner/Decision Maker	Q1: Management of student Information electronically Q3: Domain Expert: High Users: Low			Q14: ICT unit	Q12: 19/02/2018 Q13: 13/05/2018	
Requirement Engineer				Q16: System Analyst: ASLapai Developer: JDLab NG	Q11: Initiation - 02/02/2018 Execution - 19/02/2018 - 03/04/2018 Finish– 07/05/2018	
End User	Q4: Candidate Application, Robust Record Management, Candidate Profiling & Automatic Data Migration, SMS & Email Feedbacks, Smart Courseware, Seamless e-Payments, E-exam Cards Printouts, Auto-Matric No. Generation, Instant e- Receipts, Notification / Admission Letter, Live Chat and Interactive Support System Q5: Record management system, Matric number generation, Manual receipt system, Candidate application and Course registration EQ2: English	Q21: Registering student manually. EQ5: How would you like to see the information provided by the system?				Q9: To be able to manage students records in real time.

Table 5. Case Study of User Requirement Specification Base on Kipling's 5w1h Technique and Zachman Framework

domain for software requirement elicitation and to develop a software system to concretize the framework.

References

[1]. **Abad, Z. S. H., Shymka, A., Pant, S., Currie, A., & Ruhe, G. (2016, September).** What are practitioners asking about Requirements Engineering? An exploratory analysis of social Q&A sites. In *Requirements Engineering Conference Workshops (REW), IEEE International* (pp. 334-343). IEEE.

[2]. **Carmagnola, F. (2008).** The five Ws for user model interoperability. In *Ubiquitous User Modeling* (pp. 30-36).

[3]. **Chen, X., & Jin, Z. (2016).** Capturing requirements from expected interactions between software and its interactive environment: an ontology based approach. *International Journal of Software Engineering and Knowledge Engineering*, 26(01), 15-39.

[4]. **Chung, S., Won, D., Baeg, S. H., & Park, S. (2009, January).** Service-oriented reverse reengineering: 5W1H model-driven re-documentation and candidate services identification. In *Service-Oriented Computing and Applications (SOCA), 2009 IEEE International Conference on* (pp. 1-6). IEEE.

[5]. **Hadar, I., Soffer, P., & Kenzi, K. (2014).** The role of domain knowledge in requirements elicitation via interviews: An exploratory study. *Requirements Engineering*, 19(2), 143-159.

[6]. **Hanif, M. K., Talib, M. R., Haq, N. U., Mansoor, A., Sarwar, M. U., & Ayub, N. (2017).** A collaborative approach for effective requirement elicitation in oblivious client environment. *International Journal of Advanced Computer Science and Applications*, 8(6), 179-186.

[7]. **Hart, G. (1996).** The five W's: An old tool for the new task of task analysis. *Technical Communication*, 43(2), 139-145.

[8]. **Hickey, A. M., & Davis, A. M. (2003, September).**

Elicitation technique selection: How do experts do it? In *Requirements Engineering Conference, 2003. Proceedings. 11th IEEE international* (pp. 169-178). IEEE.

[9]. **Jabar, M. A., Ahmadi, R., Shafazand, M. Y., Ghani, A. A. A., & Sidi, F. (2013, August).** An automated method for requirement determination and structuring based on 5W1H elements. In *Control and System Graduate Research Colloquium (ICSGRC), 2013 IEEE 4th* (pp. 32-37). IEEE.

[10]. **Kipling, R. (1902).** *Just so Stories*. London, UK: Macmillan & Co.

[11]. **Lee, Y., & Zhao, W. (2006, June).** An ontology-based approach for domain requirements elicitation and analysis. In *Computer and Computational Sciences, 2006. IMSCCS'06. First International Multi-Symposiums on* (Vol. 2, pp. 364-371). IEEE.

[12]. **Mishra, A., Awal, A., & Elijah, J. (2017).** Automation of requirement analysis in software engineering. *International Journal on Recent and Innovation Trends in Computing and Communication*, 5(5), 1173-1188.

[13]. **Mohanan, M., & Samuel, P. (2016).** Software Requirement Elicitation using natural language processing. In *Innovations in Bio-Inspired Computing and Applications* (pp. 197-208). Springer, Cham.

[14]. **Moon, M., Yeom, K., & Chae, H. S. (2005).** An approach to developing domain requirements as a core asset based on commonality and variability analysis in a product line. *IEEE Transactions on Software Engineering*, 31(7), 551-569.

[15]. **Nuseibeh, B., & Easterbrook, S. (2000, May).** Requirements engineering: A roadmap. In *Proceedings of the Conference on the Future of Software Engineering* (pp. 35-46). ACM.

[16]. **Oranje, A. P., Al-Mutairi, N., & Shwayder, T. (Eds.). (2016).** *Practical Pediatric Dermatology: Controversies in Diagnosis and Treatment*. Springer.

[17]. **Pohl, K. (2016).** *Requirements engineering fundamentals: A study guide for the certified professional for requirements engineering exam-foundation level*. Rocky Nook, Inc.

[18]. **Reza, A. Y. (2015).** *Requirements Engineering practices in global software engineering organizations: A study in the banking industry* (Master Thesis, Delft University of Technology).

[19]. **Sadiq, M. (2017).** A fuzzy set-based approach for the prioritization of stakeholders on the basis of the importance of software requirements. *IETE Journal of Research*, 63(5), 616-629.

[20]. **Shabir, M. (2016).** Enhancing students' ability in extending ideas in the narrative writing through the 5W1H concept. *English Journal*, 17(2), 48-54.

[21]. **Sommerville, I. (2005).** Integrated requirements engineering: A tutorial. *IEEE Software*, 22(1), 16-23.

[22]. **Sommerville, I., Sawyer, P., & Viller, S. (1998, April).** Viewpoints for requirements elicitation: A practical approach. In *Requirements Engineering, 1998. Proceedings. 1998 Third International Conference on* (pp. 74-81). IEEE.

[23]. **Sutcliffe, A., & Sawyer, P. (2013, July).** Requirements elicitation: Towards the unknown unknowns. In *Requirements Engineering Conference (RE), 2013 21st IEEE International* (pp. 92-104). IEEE.

[24]. **Tiwari, S., & Rathore, S. S. (2017).** A methodology for the selection of requirement elicitation techniques. *arXiv preprint arXiv:1709.08481*.

[25]. **Wang, M., & Zeng, Y. (2009).** Asking the right questions to elicit product requirements. *International Journal of Computer Integrated Manufacturing*, 22(4), 283-298.

[26]. **Wong, L. R., Mauricio, D. S., & Rodriguez, G. D. (2017).** A systematic literature review about software requirements elicitation. *Journal of Engineering Science and Technology*, 12(2), 296-317.

[27]. **Zachman, J. A. (2003).** The Zachman framework for enterprise architecture: primer for enterprise engineering and manufacturing. *Zachman International*. Retrieved from http://www3.cis.gsu.edu/dtruex/courses/CIS8090/Cases-Articles/ZachmanBookRFIextract.pdf

[28]. **Zhaoyin, Z., Yanfang, L., & Chao, C. (2009, December).** Software requirement analysis research based on event-driven. In *Computer Science-Technology and Applications, 2009. IFCSTA'09. International Forum on* (Vol. 1, pp. 247-250). IEEE.