International Journal on Advanced Science Engineering Information Technology

Vol.10 (2020) No. 1
ISSN: 2088-5334

A Study on Reuse-based Requirements Engineering by Utilizing Knowledge Pattern

Sabrina Ahmad[#1], Zariza Hashim[*], Siti Azirah Asmai[#2]

[#]*Centre for Advanced Computing Technology, Fakulti Teknologi Maklumat Dan Komunikasi, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia*
E-mail:[1]sabrinaahmad; [2]azirah@utem.edu.my

[*]*Malaysia Maritime Academy, Batu 30, Kampung Tanjung Dahan,78200 Kuala Sungai Baru, Melaka, Malaysia*
E-mail: zarizahashim@gmail.com

***Abstract*— Software development has become an essential part of many industries over the past decade. The use of software has become an essential element for the organization to support its operation and business. Some software has certain features in common, which allow its requirements to be used repetitively in the requirement engineering phase. This paper presents a study on knowledge patterns for reuse-based requirements engineering. Reuse-based requirements engineering is saving the effort to conduct the process and, at the same time maintaining the standard since reused requirements come with its properties as well. Software development is an iterative process itself and so does the knowledge it holds in every iteration. When analysts perform many iterations of elicitation processes, it is often the case that a significant amount of requirements is recurring and similar software system will likely benefit from them. This research adopted a literature review method to investigate and to present current studies on knowledge pattern for the purpose of reuse. Knowledge reuse by utilizing knowledge pattern is becoming a significant method in software requirements engineering as it safes the effort of developing requirements from scratch. The study found that a specific pattern is required to develop good requirements specification. A proposed prototype to deploy reuse-based requirements engineering is also presented and evaluated. Experts' judgment method is used for evaluation by adapting the Technology Acceptance Model (TAM). The results showed that reusing knowledge pattern expedites the requirements elicitation process and improves the requirements quality.**

***Keywords*— requirement engineering; knowledge pattern; requirement reuse; knowledge reuse; software engineering.**

I. INTRODUCTION

It is seldom that technical aspect inhibits the software development process but usually human factors which is dominant in the process of understanding what to be built [1], [2], [3]. Brooks [4] stated that "The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirements and no other part of the work so cripples the resulting system if done wrong. No other part is as difficult to rectify later."

Therefore, reuse-based requirements engineering is saving the effort to conduct the process and at the same time maintaining the standard since reused requirements come with its properties as well. Software development is an iterative process itself and so does the knowledge it holds in every iteration. When analysts perform many iterations of elicitation processes, it is often the case that a significant amount of requirements is recurring and similar software system will likely benefit from them. [5]. In many cases, it requires an experienced analyst to recognize essential requirements patterns to make it meaningful in software development project they are working on. An experienced analyst usually able to identify lacking or missing information on user's statement while conducting requirement elicitation because of vast experience in previous projects [6]. The more project they involve means more information they could obtain. With the information, those experienced analysts may develop a library of requirements pattern in their mind without they even realized it.

This research is looking into utilizing the knowledge pattern to deploy reuse-based requirements engineering for a better quality requirements specification. The knowledge pattern makes available will provide the requirement engineers with a set of previous functional requirements to be analyzed and to reuse the requirements during the analysis phase of requirement engineering [7]. The reuse

capability also saves enormous effort to conduct a full cycle of requirements engineering process [8], [9]. In addition, the requirements that are already being used and deployed in previous projects are verified and validated to be functioning well. The effort is deemed useful because it is common knowledge that software system within the same business domain usually shares similar functional requirements characteristics.

The purpose of this paper is to present the findings of a background study on reuse-based requirements engineering by utilizing knowledge pattern. The tool-based prototype to deploy the requirements reuse is presented and evaluated through the experts' judgment technique to reveal the usefulness of reuse-based requirements engineering. Following Introduction, Section I presents related work on reuse-based requirements engineering, proposed prototype. Then, Section II describes materials and methods. Section III presents results and discussions. Section IV concludes the paper.

A. Related Works

Knowledge reuse by utilizing knowledge pattern is becoming a significant method in software requirements engineering [10]. The reuse initiative is even more significant nowadays as developing from scratch is no longer popular. This is due to the emerging trend of component-based software development [11]. Therefore, many studies have been conducted to improve requirements pattern utilization method.

It is something common that software houses usually addresses good design practice to allow reusability of the software deliverables which includes requirements statements and its' properties. However, the sustainability of the requirements are often overlooked. A research has been done to address the sustainability of software requirements which proposed sustainable requirement pattern. The pattern provides support to guide the writing of specific type of requirements. The result of study has produced a draft of sustainability requirement patterns document [12].

In relation to the availability of requirements pattern document, a study was conducted [13] to find out the attributes of design patterns that is applicable to requirements statements. The researcher explored the notations of object-oriented modeling particularly Unified Modeling Language (UML) in order to represent a collective of requirements patterns specifically for embedded systems. They used a template to facilitate the understanding of the requirement patterns and how the requirements were going to be translated into applications. The template is designed in a way to ease the understanding, which depict the problem with its context through a problem frames, and supported by the usage of models through UML diagrams to provide structural and behavioral information. Researchers conclude that these patterns are found constructive in assisting the requirements engineering process.

Moreover, requirement patterns containing basic information such as name, purpose, services and the predefined requirement template may lead to the formulation of trustworthy set of requirements patterm which can be used in system development projects. A study has been conducted [14] to find out the attributes of a trustworthy system, and the way patterns of software requirement can denote trustworthy systems. Further investigation has been done through a workshop with expert judgment method. It is observed that there exist diverse opinions regarding requirement patterns. Several experts highlighted that addresses backgrounds and requirements as important, some of the software engineers denoted that the pattern as insignificant. Further investigation has been done and discovered that trustworthy towards software system can be improved by fulfilling insignificant requirements, with a condition that the users are part of the development project. The study concludes that in relation of trustworthy pattern among end users, usability is highly involved.

Other than requirements specification, activity in product lines can be reuse as well [15], [16]. The reuse effort at this stage helped to systematically estimate alternative options through reuse scenarios in evaluating and comparing effective support of the make/buy process. The model to systematically identify reuse options is able to differentiate between the reuse services, which replicated artifacts and maintenance attributes that are concerned with making modifications to artifacts within the same domain. In their study, the researchers described and demonstrated how the model is used to realized systematic trustworthy pattern reuse. A conceptual model with the help of a case study is used to demonstrate the realization of requirements reuse in practice. The study concluded that their model is able to produced clear identification of basic services needed together with the related properties which also embedded with cost component in a focused and accurate way.

Beyond knowledge pattern to promote reusability, a study has been conducted to utilize software metrices to allow repository traceabilty in order to utilize knowledge reuse. The researcher [17] established a software structure which make the connections between metric and reuse traceability possible, with reuse library metrics. A framework was then established to allow access to the value of reuse program within an organization.

Various studies have been carried out on knowledge pattern and PABRE appeared as the most detail framework in constructing and managing requirements pattern catalogue [13]. PABRE framework adopts software requirements patterns (SRP) as an approach to reuse. The SRP is utilized within the PABRE framework as a way to get and to use requirements knowledge in the context of IT procurement projects. The framework is realized through a tool system called PABRE-Man (pattern management tool) to facilitate the definition of SRP and their organization into an SRP catalogue. Furthermore, PABRE-Proj is developed to enable the requirements definition for a software project. It helps to recommend, using the SRP catalogue, the requirements to add to the Software Requirements Specification (SRS) document of the project [5].

B. The Prototype

This section describes a tool based prototype to deploy reuse-based requirements engineering in order to export the knowledge into the requirements document [18]. The knowledge pattern anatomy is inspired from [19].

1) The Flow

Figure 1 shows the flow of the proposed tool based prototype to implement reuse by utilizing the knowledge pattern. The process starts with the requirements engineer searching the requirements based on keywords. Then, the tool will display the searching results in a form of a list of available requirements that match the searching criteria. The requirements engineer will then view and compare the requirements available to be chosen from. Subsequently, they can choose the requirements and export the requirements into word processing tools such as Microsoft Word. The imported requirements can be edited to fit in the new software development project and included in the requirements document. If the requirements being searched is not available, the requirements engineer are allowed to key in a new requirement. The new requirements will be then saved in the repository knowledge for future use.

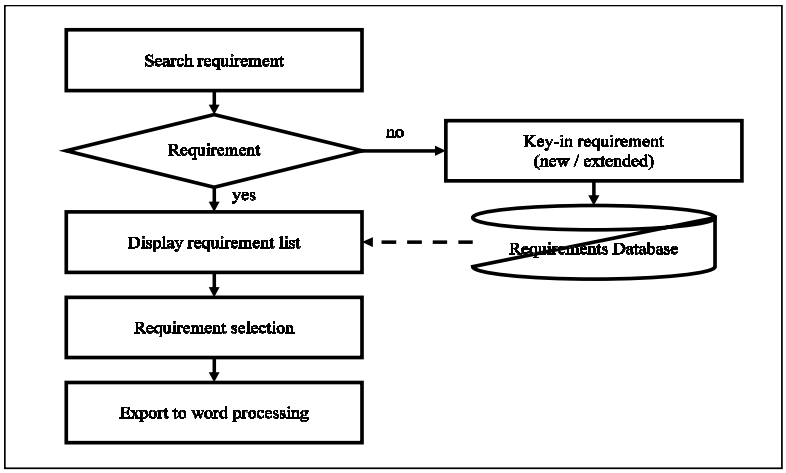

Fig. 1 The Prototype Flow

2) The Prototype Screens

The prototype consists of two screens. The first screen enables users to search, view and export the requirements to be reused. The suitable and selected requirements can be exported into the word processing application. The second screen enables the user to add new requirements that potentially form new requirements pattern.

Figure 2 illustrates the first screen which allows requirements engineers to view a list of available requirements for reuse. This screen will assist the requirements engineer to select a suitable requirement by showing the percentage similarity of requirements according to their search. The best option will be then exported into the word processing application.

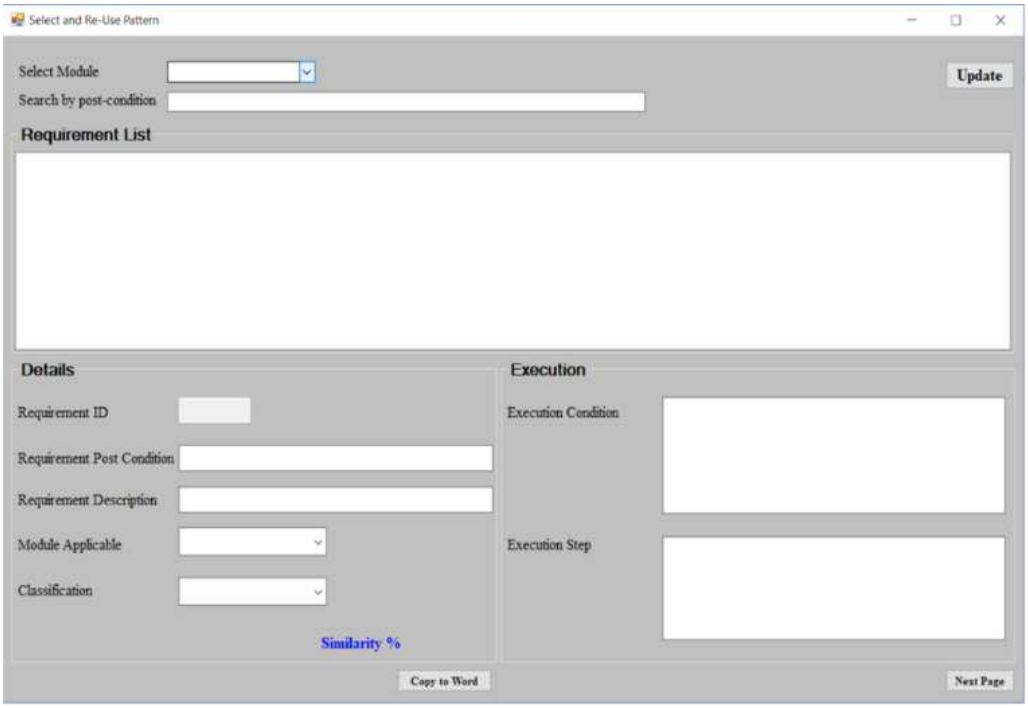

Fig. 2 Screen 1

The flow of event:

[1] Select module, list of requirements appears in the requirement list section.
[2] Search requirement by post-condition (objective).
[3] Check the requirement by clicking on the list of requirements.
[4] Display the similarity percentage of the selected requirement by comparing the post-condition search.
[5] Once the suitable requirement is found, click on the 'Copy to word' button to export selected requirements into Microsoft word application.

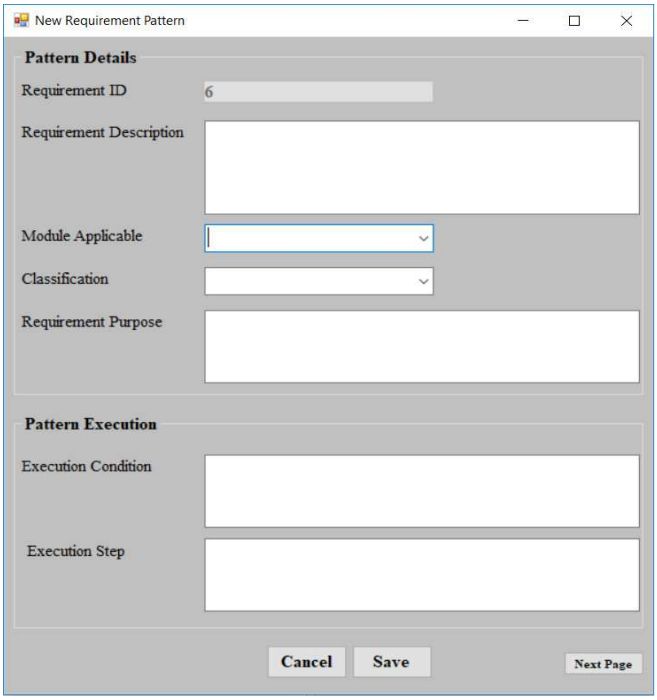

Fig. 3 Screen 2

Figure 3 illustrates the second screen, which allows the requirements engineer to enter new requirements to be added into the database and therefore being part of the growing knowledge. The flow of the event is as follows:

[6] Key-in requirements details.
[7] Save records into the database.

II. MATERIALS AND METHODS

This section describes the procedure to evaluate the reuse-based requirements engineering effort through a prototype to utilize knowledge patterns in order to improve requirements elicitation performance. The expert judgment method is used for the evaluation. The foundation of the method is the translation of experts' tacit knowledge into probabilistic measures associated with the achievement level of improving the requirements elicitation process with quality. Through the questionnaire, experts' preferences were captured to be the foundation for an aggregated quality measure.

A. Identifying Experts

Five experts were identified to participate in the evaluation. They were practitioners dealing with requirements engineering in the Malaysian industry. Their experience ranges from five to ten years with a different level of seniority. The experts were also formally trained in software engineering good practice with at least a first degree. Besides, all the experts are involved and familiar with the requirements elicitation and analysis process in their current employment. Figure 4 until 6 summarizes the demographics of the experts.

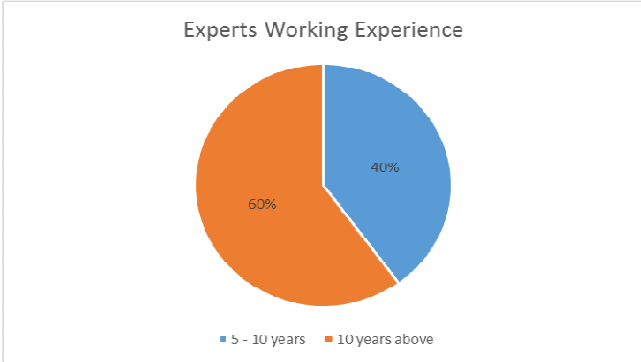

Fig. 4 Experts Working Experience

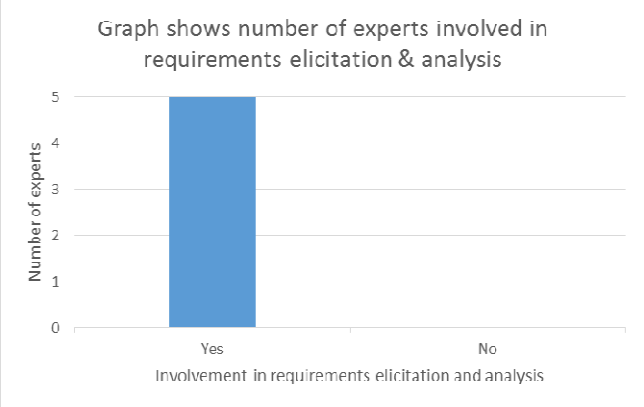

Fig. 5 Experts involvement in requirements elicitation and analysis in current employment

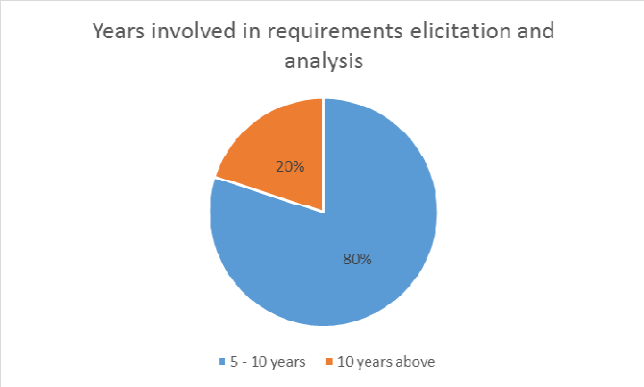

Fig. 6 Years involved in requirements elicitation analysis

B. Instruments

The experts are the instrument for this evaluation protocol as they will provide their expert judgment to the effectiveness of knowledge pattern reuse during the requirements elicitation process. The prototype is used to allow the experts to experience the reuse-based requirements engineering and the questionnaire is designed to capture the experts 'opinion. The questionnaire was designed based on

the understanding of the Technology Acceptance Model (TAM) and the elements evaluated are perceived usefulness and perceived ease of use, which influence the attitude towards using the reuse knowledge pattern during requirements elicitation process.

According to the TAM, a potential user's overall attitude toward using a particular system or an application is hypothesized to be a significant determinant of whether or not he actually uses it. Attitude toward using, in sequence, is a function of two significant beliefs, which are perceived usefulness and perceived ease of use. Perceived usefulness is defined as "the degree to which an individual believes that using a particular system would enhance his or her job performance" while perceived ease of use is defined as "the degree to which an individual believes that using a particular system would be free of physical and mental effort." [20].

Perceived ease of use is believed to have a causal effect on perceived usefulness since, all else being equal, a system that is easier to use will result in increased job performance. Design features usually influence both perceived usefulness and perceived ease of use. They are not theorized to have any direct effect on attitude or behavior, instead of affecting these variables only indirectly through perceived usefulness and perceived ease of use[20]. Therefore, the likelihood of perceived usefulness and perceived ease of use to influence the attitude to use the pattern tool application in order to improve performance and quality is high. Table I describes questions in the questionnaire.

TABLE I
TECHNOLOGY ACCEPTANCE MODEL (TAM) INSPIRED QUESTIONNAIRE

(TAM) Elements	Questions
Perceived Usefulness	1. Do you agree that having a pattern tool application can assist in reuse and requirements management? 2. Do you agree that pattern tool application can make it easier to elicit requirements? 3. Do you agree that pattern tool application can help in producing good quality requirements? 4. Do you agree that the pattern tool application can help in improving the requirements elicitation performance?
Perceived Ease of Use	1. The pattern tool application is easy to use. 2. Learning how to use a pattern tool application is easy for me. 3. It is easy to become skillful at using the pattern tool application.
Attitude	1. Using pattern tool application expedite the requirements elicitation process. 2. Using pattern tool application improves requirements quality.

C. The Protocol

The evaluation protocol is divided into two parts. The first part requires experts to use the tool prototype in order to experience the reuse-based requirements engineering. Beforehand, the researcher provides a briefing to the expert and demo the tool prototype flow and functionalities. Then, ample time is given to the experts to experience the reuse-based requirements engineering through the tool prototype. The second part requires the experts to answer the questionnaire as described in Table 4 in order to capture the experts' judgment.

III. RESULTS AND DISCUSSION

All five experts' judgments were gathered and analyzed. The judgment was about if the reuse-based requirements engineering improves the requirements elicitation process. The implementation of the reuse approach was made easy with the assistance of the tool which is referred to as a pattern tool application in this paper.

An analysis of Perceived Usefulness (PU), which derived from four questions, showed that the majority of the experts agreed that reuse-based requirements engineering through the pattern tool application are useful, as illustrated in Figure 7.

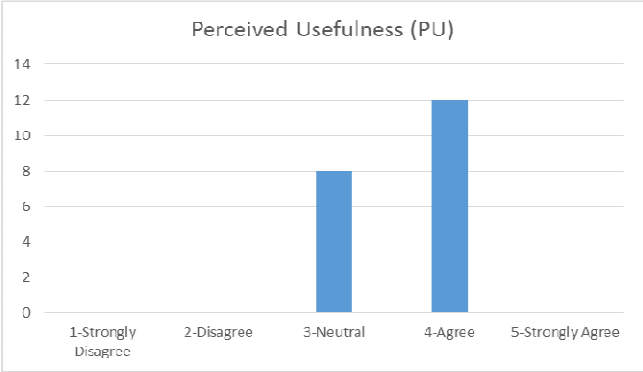

Fig. 7 Perceived Usefulness (PU)

Besides, an analysis of the Perceived Ease of Use (PEoU) which derived from three questions, showed that the experts agreed that the pattern tool application is easy to be utilized while performing requirements elicitation process with reuse, as illustrated in Figure 8.

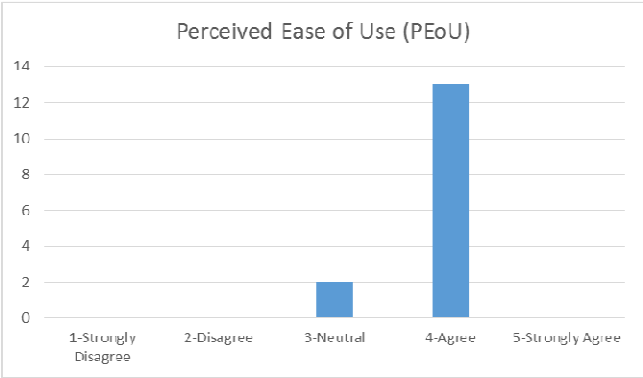

Fig. 8 Perceived Ease of Use (PEoU)

The evaluation results confirm that the theory saying PU and PEoU influence the Attitude (A) is valid. Figure 9 shows that the majority of the experts agreed that reuse-based requirements engineering assists in expediting the requirements elicitation process and improves requirements quality.

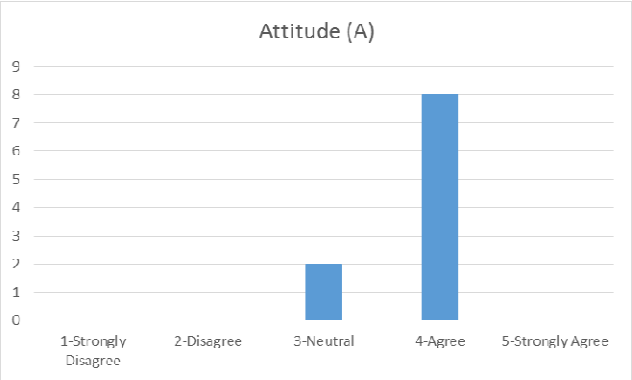

Fig. 9 Attitude (A)

IV. CONCLUSIONS

This paper provides background knowledge of reuse requirements engineering by utilizing knowledge patterns. Related researches are discussed, and the efforts lead to the development of the prototype tool to deploy reuse requirements engineering. The prototype evaluation through experts' judgment method by adapting the Technology Acceptance Model (TAM) shows the experts agreed that reuse improves requirements elicitation process performance and quality.

ACKNOWLEDGMENT

Universiti Teknikal Malaysia Melaka funded the publication of this paper through a research grant numbered PJP/2017/FTMK-CACT/S01573.

REFERENCES

[1] U. Anuar, S. Ahmad, N.A. Emran, "A simplified systematic literature review: Improving Software Requirements Specification quality with boilerplates" *Proceedings of 9th Software Engineering Conference (MySEC)*, pp. 99-105 IEEE, 2015.

[2] S. Ahmad, S. A. Asmai, "Measuring software requirements quality following negotiation through empirical study," *International Journal of Applied Engineering Research*, vol.11, no. 6, pp. 4190-4196, 2016.

[3] S. Ahmad, I.E.A. Jalil, S.S. Syed Ahmad, "An Enhancement of Software Requirements Negotiation with Rule-based Reasoning: A Conceptual Model," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 8, no. 10 pp. 193-198, 2016.

[4] F. Brooks, "The mythical man-month: essays on software engineering," *Choice Reviews Online*, vol. 33, no. 09, pp. 33-5142-33–5142, May 1996.

[5] C. Palomares, C. Quer, and X. Franch. "Requirements reuse and requirement patterns: a state of the practice survey." *Empirical Software Engineering* 22, no. 6 (2017): 2719-2762

[6] M. Irshad, K. Petersen, and S. Poulding, "A systematic literature review of software requirements reuse approaches," *Inf. Softw. Technol.*, vol. 93, pp. 223–245, Jan. 2018.

[7] B. Hamid and J. Perez, "Supporting pattern-based dependability engineering via model-driven development: Approach, tool-support and empirical validation," *Journal of Systems and Software*, vol. 122, pp. 239–273, Dec. 2016.

[8] S. Schacht, and A. Maedche. "A methodology for systematic project knowledge reuse." *In Innovations in Knowledge Management*, pp. 19-44. Springer, Berlin, Heidelberg, 2016.

[9] D. Bombonatti, M. Goulão, and A. Moreira, "Synergies and tradeoffs in software reuse - a systematic mapping study," *Softw: Pract. Exper.*, vol. 47, no. 7, pp. 943–957, Jul. 2017.

[10] T. Diamantopoulos and A. Symeonidis, "Enhancing requirements reusability through semantic modeling and data mining techniques," *Enterprise Information Systems*, vol. 12, pp. 1–22, Dec. 2017.

[11] M. Tahir, F. Khan, M. Babar, F. Arif, S. Khan, "Framework for Better Reusability in Component-Based Software Engineering" *Journal of Applied Environmental and Biological Sciences*, vol. 6, no. 4, pp. 77-81, 2016.

[12] C. C. Venters, N. Seyff, C. Becker, S. Betz, R. Chitchyan, L. Duboc, D. McIntyre, and B. Penzenstadler. "Characterising sustainability requirements: A new species red herring or just an odd fish?." *In 2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Society Track (ICSE-SEIS)*, pp. 3-12. IEEE, 2017.

[13] P. Filipovikj, "Pattern-Based Specification and Formal Analysis of Embedded Systems Requirements and Behavioral Models." 2017 http://www.es.mdh.se/pdf_publications/4923.pdf.

[14] A. Hoffmann, M. Söllner, H. Hoffmann, and J. M. Leimeister. "Towards Trust-Based Software Requirement Patterns," 2012.

[15] J. L. Barros-Justo, F. B. V. Benitti, and A. L. Cravero-Leal, "Software patterns and requirements engineering activities in real-world settings: A systematic mapping study," *Computer Standards & Interfaces*, vol. 58, pp. 23–42, May 2018.

[16] R.L.Q. Portugal, and J. C. S. do Prado Leite. "Extracting requirements patterns from software repositories." *In 2016 IEEE 24th International Requirements Engineering Conference Workshops (REW)*, pp. 304-307. IEEE, 2016.

[17] J. C. Cordeiro Pires Mascena, E. Santana de Almeida, and S. R. de Lemos Meira, "A comparative study on software reuse metrics and economic models from a traceability perspective," in *IRI -2005 IEEE International Conference on Information Reuse and Integration, Conf, 2005.*, 2005, pp. 72–77.

[18] V. Chang, M. Abdel-Basset, and M. Ramachandran, "Towards a Reuse Strategic Decision Pattern Framework – from Theories to Practices," *Inf. Syst. Front.*, pp. 1–18, May 2018.

[19] S. Withall, "Software Requirement Patterns" 2007.

[20] L. M. Maruping, H. Bala, V. Venkatesh, and S. A. Brown, "Going beyond intention: Integrating behavioral expectation into the unified theory of acceptance and use of technology," *J. Assoc. Inf. Sci. Technol.*, vol. 68, no. 3, pp. 623–637, May 2016