Requirements Engineering Practices in UUMIT Centre: An Assessment Based on the Perceptions of In-House Software Developers

Azham Hussain, Emmanuel O.C. Mkpojiogu, Inam Abdullah
School of Computing, Universiti Utara Malaysia, 06010 Sintok, Malaysia
azham.h@uum.edu.my

***Abstract*—Requirements Engineering (RE) is a systematic procedure that entails and encompasses the elicitation, elaboration, documentation, negotiation, validation and management of the system's requirements in a software engineering project. Universiti Utara Malaysia (UUM) is been supported by several systems, engineered by the UUM Information Technology (UUMIT) Centre. The objective of this paper was to investigate the requirements engineering practices at UUMIT Centre. The major issue that led to this study was the absence of studies that support software development efforts at the UUMIT Centre. This research is aimed at assisting UUMIT Centre in developing quality, and as well, time and cost saving software systems through the employment of state of the art requirements engineering practices. Furthermore, the paper, as a contribution to UUM, identifies the activities that are needed for software construction to enable the University management allocate budget for the provision of adequate and cutting edge training for the in-house software developers. Three variables were assessed: Requirement Description, Requirements Development (consisting of: Requirements Elicitation, Requirements Analysis and Negotiation, Requirements Validation), and Requirement Management. The results from this research revealed that the current practices of requirement engineering in UUMIT is good and commendable, however there is need and room for more improvement in a few RE practices that were rarely practiced. In addition, recommendations were also proffered for effective training programs for UUMIT staff on RE practices to build the capacity of in-house developers and other associated staff. The training will increase their understanding on system requirements using RE practices to enable them develops better systems for the university. Further investigation is required in the future to understand the effect of RE practices on software development. In addition, also as a future work, the researchers aim to extend the scope of this study to other government and non-educational organizations.**

***Index Terms*—Requirements Engineering Practices; Requirements Description; Development and Management.**

I. INTRODUCTION

Nowadays, software applications have considerably supported our work and daily life. Software applications are everywhere. Certainly, there is a dire need to develop software that satisfies the needs of the users without any error or at least with very minimal error. Requirements of software are captured through requirements engineering (RE) which is the process of determining requirements [1]. Cheng and Atlee [1] mentioned that successful Requirements Engineering (RE) involves the discovering of the stakeholders needs, understanding of the requirements contexts, modelling, analyzing, negotiating, validating, as well as assessing documented requirements; and managing of the requirements [6]. There are many studies that identify the need for the development of quality software that meet the needs and objectives of the customers and give value to stakeholders [2-4], [38-46]. Asghar and Umar [5] pointed out that RE is acknowledged as the first phase of software engineering process and it is considered as one of the main phase in software development. Furthermore, Khan *et al.* [2] and Shah and Patel [6], asserted that, unclear requirement is the main reason of software project failure. Khan et al. [2] said that "requirement engineering phase is difficult and crucial". Also, Young [7] stated that the neglect of RE contributes to project failures. Requirement engineering impacts productivity as well as product quality [39]. Thus, it can be stated that RE is an essential phase for software development [8], and therefore RE practices should be taken into consideration in every software development project [38].

In this study, RE process is defined based on as proposed by Wiegers [29]. He maintained that RE is composed of two main activities which are: requirements development and requirements management. Thus, this study focuses on these two activities. According to Kavitha and Thomas [9], proper comprehension and management of requirements are the main determinants of success in the process of development of software. UUM is supported by various systems. Most of these systems are developed and maintained by UUMIT. Thus, it is important for UUMIT centre to deliver quality software in time and within budget. Requirements engineering can support organizations in developing software systems that are of quality, in time and within budget and that reflect the true needs of the customers [10]. UUMIT has its own software developers that develop software in-house to support the business functions of UUM. This study investigates how the software developers at UUMIT practice RE during software development. In software development, a project is considered successful whenever it is able to deliver within the time frame and budget and the developed system has all the specified features and functions. The main reasons for software project failure borders on requirements (e.g. poor, changing, ambiguous or incomplete requirements). Poor specification leads developers to making incorrect business logic decision

[4], [11-12]. Inadequate requirements contribute 73% to project failure rate [13]. Lindquist [14] reported that 71% of software projects failure is due to poor requirements. This makes it the single biggest reason for project failure. In order to promote software project success, RE plays an active and vigorous role in system development project [15]. Kumar and Kumar [16] stated that poor requirements lead to increase in the overall project cost, decrease in quality of the system or project failure altogether.

II. Requirements Engineering Practices

Every project has some basic requirements that define what the end users, clients, customer, developers, suppliers or business (i.e. stakeholders) require from it coupled with some need of the systems for efficient functioning. Requirement is a key factor during every software development as it describes what different stakeholders need and how the system will satisfy these needs. It is generally expressed in natural language so that everyone can understand it well. It helps the analyst to better understand which element and function are necessary in the development of a particular project. More so, requirements are considered as an input to design, implementation and validation phases of software product development. Thus, a software project is successful or a failure during software development because of the state of requirement elicitation as well as that of the requirements management process. According to Pfleeger and Atlee [18], requirements are categorized as functional, non-functional requirements and constrains. The 1995 Chaos report established that RE practices contributed more than 42% of overall project success. Likewise, inappropriate RE practices represent more than 43% of the reasons for software projects failure. In addition, many previous researchers have identified that 70% of the requirements were difficult to identify and 54% were not clear and well organized [21-22]. Gause and Weinberg [22] also pointed out that: i) Requirements are difficult and challenging to describe in natural language; ii) Requirements have many different types and levels of details; iii) Requirements are difficult to manage if they are not in control; iv) Most of the requirements, change during software development.

In 2004, [30] conducting a study in Australian organizations, pointed out several key factors that influence the success of requirements management in software development structure. In addition, a study by [31] (who conducted a survey at the Vietnamese software industry) concluded that cultural issues are responsible in maintaining trust in outsourcing software. Further surveys conducted by [32] and [33], concentrate on requirements expert. Their surveys focused on specific problem rather than on understanding the general industry problem. In 2010, [34] performed an empirical study of RE in Chinese companies. In this study, they reported that the neglect of RE leads to project failure in the industry and they provided the reasons for the failure. In the Malaysian software industry, a recent study was conducted in 2014 by [35]. In this study, they focused on the RE practice in Malaysia public sector. This study reported that the main problem was communication between system analysts and stakeholders.

III. Methodology

A survey was conducted at UUMIT centre. The aim and purpose of this survey was to investigate how the software developers at the UUMIT conduct their RE practices presently. The objectives of this survey were to investigate: i) how the software developers describe their requirements; ii) how the software developers conduct the requirements development during software development; iii) how the software developers conduct the requirements management during software development. A self-administered questionnaire was used to collect the data regarding current RE practices at UUMIT. The questionnaire was adapted from Iqbal et al. [17], and Khankaew and Riddle [36]. The adapted questionnaire consists of 49 questions that capture the requirements engineering practices of developers at the UUMIT. The items were measured on a likert-type scale (with five options: never, rarely, sometimes, regularly and always), as adapted from Zainol and Mansoor [37]. The data collected was analyzed using SPSS version 20 software package. Data were collected from 20 participants. Descriptive statistics (simple percentages) was used in the analysis. The following variables were captured: i) Requirements Description; ii) Requirements Development, (which compose of the following variables: Requirements Elicitation; Requirements Analysis and Negotiation; Requirements Validation); and iii) Requirement Management.

IV. Results and Discussion

The results of the study show a moderate practice of requirements engineering best practices among the software developers at UUMIT.

Table 1
Frequency of Requirements Description Practices

Requirements Description Practices	Often (%)	Rarely (%)
Have standards templates/documents for describing requirements	85	10
Have a specific lay out for the requirements document to improve readability	80	15
Have guidelines on how to write requirements	75	20
Produce a summary of the requirements	80	15
Make a business case for a system	60	35
Have a glossary of specialized terms	55	40
Requirements document easy to change	55	45
Use diagrams appropriately	55	40
Supplement natural language with other descriptions of requirements	35	60
Specify requirements quantitatively	50	45
Requirements Description (Average)	63	32.5

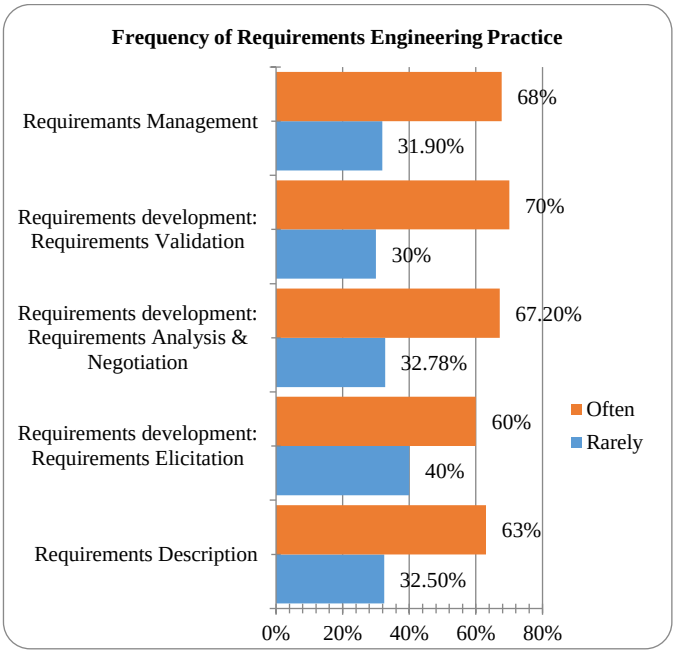

Figure 1: Frequency of requirements engineering practice in UUMIT

A. Requirement Description

It was found that software developers frequently practice the use of standard templates/documents for describing requirements (85%) and they have specific layout for the requirements document to improve readability (80%). They often produce summary of requirements (80%) and use guidelines to write requirements (75%) (Table 1). The study showed that currently there is a weak practice in supplementing natural language with other descriptions of requirements (35%) (Table 1). In summary the developers practiced the following steps of requirement description on regular bases: Using guidelines to write requirements, producing a summary of the requirements, making a business case for a system, defining a glossary of specialized terms, using diagrams appropriately, and specifying requirements quantitatively. Most of the developers indicated that their practice of requirement description were frequent (63%). 32.5% indicated that theirs were rare (see Figure 1).

B. Requirements Development (RD)

Three variables that make up requirements development (RD) were examined: Requirements Elicitation, Requirements Analysis and Negotiation, and Requirements Validation. The following is the summary:

C. Requirements Elicitation (RE)

The result of the study showed that the majority of participants regularly practice requirements elicitation (60%); (those that rarely practice is 40%); (see Figure 1), developers always carry out feasibility study before starting a new project in comparison to other practices (80%). They often use business concerns to drive requirements elicitation (80%) (Table 2). The weakest practice of requirement elicitation is in defining operational processes (32%) (Table 2). In summary, the elicitation practice was carried out in the following ways: Staff were sensible to organizational and political factors which influence requirements sources, using business concerns to drive requirements elicitation, using scenarios to elicit requirements, reusing requirements from other systems which have been developed in the same application area.

Table 2
Frequency of Requirements Elicitation Practices

Requirements Elicitation Practices	Often (%)	Rarely (%)
Carry out feasibility study before starting a new project	80	20
Sensitivity to organizational and political factors while eliciting requirements	50	50
Use business concerns to drive requirements elicitation	80	20
Use scenarios to elicit requirements	55	45
Define operational processes	32	65
Reuse requirements from other systems which have been developed in the same application area	60	40
Requirements Elicitation (Average)	60	40

Table 3
Frequency of Requirements Analysis and Negotiation Practices

Requirements Analysis & Negotiation Practices	Often (%)	Rarely (%)
Define system boundaries	85	15
Use checklists for requirements analysis	70	30
Encourage the use of electronic systems (e.g., e-mail) to support requirements negotiations	60	40
Plan for conflicts and conflict resolution	70	30
Prioritize requirements	80	20
Classify requirements using a multidimensional	65	35
Approach which identifies specific types	50	50
Use interaction matrices to find conflicts and overlaps	65	35
Perform any risk analysis on requirements	60	40
Requirement Analysis & Negotiation (Average)	67.2	32.8

D. Requirements Analysis and Negotiation

The results of the study show that the majority of participants regularly practice requirements analysis and negotiation as part of requirement development (67.2%); (those that rarely practice are 32.78%); (see Figure 1). The prioritization of requirements (80%) and defining system boundaries (85%) were the two most practiced RE practices (Table 3). The following requirements analysis and negotiation practices are practiced on regular bases: Checking lists for requirements analysis, planning for conflicts and conflict resolution, classifying requirements using a multidimensional approach which identifies specific types, using interaction matrices to find conflicts and overlaps, and performing any risk analysis on requirements.

E. Requirements Validation (RV)

The majority of developers confirm that they carry out the steps of requirements validation on regular bases (70% on the average) (those that practiced rarely are 30% on the average) (see Figure 1). Developers often check requirements document to verify that they meet project standards (70%); they also often

define validation checklists in order to focus the validation process (85%) (Table 4). Other practices of requirement validations found satisfactory and achieved by participants on regular bases include: Organizing formal requirements inspections (70%), using multi-disciplinary teams to review requirements (55%), involving external reviewers (from the project) in the validation process (55%), using prototyping to animate/demonstrate requirements for validation (75%), proposing requirements test cases (75%), and allowing different stakeholders to participate in requirements validation (75%)

Table 4
Frequency of Requirements Validation Practices

Requirements Validation Practices	Often (%)	Rarely (%)
Check that requirements document meets your standards	70	30
Organize formal requirements inspections	70	30
Use multi-disciplinary teams to review requirements	55	45
Involve external (from the project) reviewers in the validation process	55	45
Define validation checklists in order to focus the validation process	85	15
Use prototyping to animate/demonstrate requirements for validation	75	25
Propose requirements test cases	75	25
Allow different stakeholders to participate in requirements validation	75	25
Requirements Validation (Average)	70	30

Table 5
Frequency of Requirements Management Practices

Requirements Management Practices	Often (%)	Rarely (%)
Uniquely identify each requirement	80	20
Have defined policies for requirements management	75	25
Record requirements traceability from original sources	85	15
Define traceability policies	60	40
Maintain traceability manual	80	20
Use a database to manage requirements	45	55
Define change management policies	60	35
Identify global system requirements	50	50
Identify volatile requirements	65	35
Record rejected requirements	65	35
Reuse requirements over different projects	80	20
Requirement Management (Average)	67.7	31.9

F. Requirements Management (RM)

The majority of developers stated that they practiced requirements management and that they do this always (67.7%), however, 31.9%, practised it rarely; (see Figure 1). The requirements management practice, practiced frequently are shown inter alia: Uniquely identifying each requirement (80%), having defined policies for requirements management (75%), recording requirements traceability from original sources (85%), reusing requirements over different projects (80%), maintaining traceability manual (80%) (Table 5). This result is encouraging and shows that the developers at UUMIT have high level of awareness on the importance of requirement management on IT project as part of overall RE practices. Other practices associated with requirement management found satisfactory and achieved by developers on regular bases are: Defining traceability policies, using a database to manage requirements, defining change management policies, identifying global system requirements, identifying volatile requirements, and recording rejected requirements.

V. RECOMMENDATIONS

Based on the results and findings of this study, the research sets forth the following recommendation: i) The staff in the IT department need to get good requirements and effectively manage those requirements as a strong predictor of project success in software development; ii) There is the need for training staff to carry out all requirements practices on regular bases to increase the rate of practices to higher level; iii) Software development methodologies that include RE processes and that lead to better results should be encouraged and used; iv) The understanding of system requirements is critical to developing good systems for the university , and thus, should be a priority; v) The UUMIT staff must give the same attention to all requirements in requirement development and requirement management and practice all requirements practices always to improve the RE process; vi) The UUMIT staff can benefit from Agile to achieve software projects with high accuracy and in shorter time with less setbacks and errors during the implementation and development of software projects. Agile working software is the principal measure of progress in RE; vii) other lean methodologies should be utilized; viii) RE in UUM need sustainable development to maintain a constant pace in software development.

VI. CONCLUSIONS, LIMITATION AND FUTURE WORK

Requirements Engineering (RE) is a systemic and integrated process of eliciting, elaborating, negotiating, validating and managing of the requirements of a system in a software development project. UUM has been supported by various systems developed and maintained by the UUM Information Technology (UUMIT) Centre. The aim of this study was to assess the current requirements engineering practices at UUMIT. The main problem that prompted this research is the lack of studies that support software development activities at the UUMIT. The study was geared at helping UUMIT produce quality but time and cost saving software products by implementing cutting edge and state of the art requirements engineering practices. Also, the study contributes to UUM by identifying the activities needed for software development so that the management will be able to allocate budget to provide adequate and precise training for the software developers. Three variables were investigated: Requirement Description, Requirements Development (comprising: Requirements Elicitation, Requirements Analysis and Negotiation, Requirements Validation), and Requirement Management.

This study analyzed and evaluated the Requirements Engineering Practices among Software Developers at UUMIT. Requirement engineering composed of two main activities, which are requirements development and requirements management. These two phases were investigated. The study investigated how the software developers at UUMIT practice the RE during software development. Therefore, it focused on how requirements were being elicited, analyzed, negotiated and validated in requirements development. With regard to requirements management activity, this study focused on how changes in the requirements, version control of requirements, traceability and tracking of requirements were handled. The results from the survey showed that the current practice of requirement engineering in IT department of UUM is encouraging but needs further enhancement because few of the RE practices associated with requirement development and requirement management were not carried out frequently (e.g. in requirement description: the frequency of the practice of supplementing natural language with other descriptions of requirement is 35% and in requirement elicitation: the frequency of the practice of defining operational processes is 32%).

Therefore, there is a need for more to be done on these practices. This can be achieved by encouraging the IT staff to practice them regularly and providing developers and allied staff training, to increase their capacity and performance in these RE practices. As afore mentioned, the study investigated only three variables: Requirements Description, Requirements Development, and Requirement Management. In addition, the study is limited to UUMIT. The researchers suggest in future studies that the same variables analyzing RE practices can be used for Small and Medium Enterprises, large corporation, government agencies and ministries. Although the findings of this study show the importance of Requirements Description, Requirements Development, and Requirement Management on RE practice in UUMIT, further investigation is required in the future to confirm and verify the results of this study by direct observation methods. As a future work, the researchers aim to extend the scope of this study to other government and non-educational organizations.

REFERENCES

[1] Cheng, B.H.C. and Atlee, J.M. "Current and future research directions in requirements engineering". *Design Requirements Engineering: A Ten-Year Perspective* 2009. Springer.

[2] Khan, H.H., and Naz'ri bin Mahrin, M. "Generating Risks during Requirement Eng. Process in Global Software Development Environment". *International Journal of Digital Information & Wireless Communication (IJDIWC). Vol.* 4(1), pp. 63-78, 2014.

[3] Wiegers, K. *Creating a Software Engineering Culture.* Addison-Wesley, 2013.

[4] Sheikh, J.A., Dar, H.S., and Sheikh, F.J. *Usability Guidelines for Designing Knowledge Base in Rural Areas Design, User Experience, and Usability. User Experience Design for Everyday Life Applications and Services.* Springer, 2014.

[5] Asghar, S. and Umar, M. "Requirement engineering challenges in development of software applications and selection of customer-off-the-shelf (COTS) components". *International Journal of Software Engineering.* Vol. 1(1), pp. 32-50, 2010.

[6] Shah, T. and Patel, V.S. "A Review of Requirement Engineering Issues and Challenges in Various Software Development Methods". *International Journal of Computer Applications.* 99(15): 36-45, 2014.

[7] Young, R.R. "*The requirements engineering handbook*. Artech House, 2004.

[8] Sankhwar, S., Singh, V. and Pandey, D. "Requirement Engineering Paradigm". *Global Journal of Multidisciplinary Studies. pp*. 3(3), 2014.

[9] Kavitha, C.R. and Thomas, S.M. "Requirement gathering for small projects using agile methods". *IJCA Special Issue on Computational Science-New Dimensions &* Perspectives. vol. 3, pp.122-128, 2011.

[10] Nilofer, M. and Sheetal, G. "Comparison of Various Elicitation Techniques & Requirement Prioritization Techniques". *International Journal of Engineering Research & Technology. Vol.* 1(3), pp. 1-8, 2012.

[11] Liu, J.Y.C., Hun-Gee, C.C.C. and Sheu, T.S. "Relationships among interpersonal conflict, requirements uncertainty, and software project performance". *International Journal of Project Management.* Vol. 29(5), pp. 547-556, 2011.

[12] Bashir, M.S. and Qureshi, M.R.J. "Hybrid Software Development Approach for Small to Medium Scale Projects: Rup, Xp & Scrum. *Cell*. 966, 536474921, 2012.

[13] Cerpa, N. and Verner, J.M. "Why did your project fail?" *Communications of the ACM 2009. Vol.* 52(12), pp. 130-134, 2009.

[14] Lindquist, C. 2005. Fixing the requirements mess. *CIO Magazine, 2005, pp.* 52-60.

[15] Ramingwong, L. "A Review of Requirements Engineering Processes, Problems & Models". *International Journal of Engineering Science & Technology. Vol*. 4(6), 2012.

[16] Kumar, S.A. and Kumar, T.A. "Study the Impact of Requirements Management Characteristics in Global Software Development Project: An Ontology Based Approach". *International Journal of Software Engineering and Application. Vol.* 2(4), 2011.

[17] Iqbal, J., Ahmad, R., Nizam, M.H., Nasir, M.D. and Noor, M.A. "Significant Requirements Engineering Practices for Software Development Outsourcing". *Software Engineering Conf. (ASWEC)* Australia, 2013.

[18] Pfleeger, S.L. and Atlee, J.M. "*Software Engineering: Theory and Practice*". 2006, India: Pearson.

[19] Zave, P. and Jackson, M. "Four dark corners of requirements engineering". *ACM Transaction on Software Engineering and Method. Vol.* 6(1), pp. 1-30, 1997.

[20] Brooks, F.P. and Bullet, S. "Essence & accidents of software engineering". *IEEE Computers. Vol.* 20(4), pp. 10-19, 1987.

[21] Leffingwell, D. and Widrig, D. *Managing software requirements: a use case approach*. Pearson, 2003.

[22] Gause, D.C. and Weinberg, G.M. "*Exploring requirements: quality before design*. 1989, New York: Dorset House.

[23] Boehm, W.B. and Papaccio, N.P. "Understanding and Controlling Software Cost". *IEEE Transaction on Software Engineering.* pp. 1462–1476, 1988.

[24] Leffingwell, D. "Calculating your return on investment from more effective requirements management". *American Programmer. Vol.* 10(4), pp. 13-16, 1997.

[25] Quispe, A., Marques, M., Silvestre, L., Ochoa, S.F. and Robbes, R. "Requirements engineering practices in very small software enterprises: A diagnostic study". International Conference of the Chilean Computer Science Society (SCCC) 2010.

[26] Leffingwell, D. and Widrig, D. *Managing software requirements: a unified approach.* Addison-Wesley Professional. 2000.

[27] Bell, D., Morrey, I. and Pugh, J. *The essence of program design.* India: Pearson Education, 1997.

[28] Pressman, R.S. Software Eng.: A Practitioner's Approach, 7/e, 2010, Pressman & Associates, 2010.

[29] Wiegers, K.E. Software Requirements: Practical techniques for gathering & managing requirement through the product development cycle. *Microsoft Corporation*. 2003.

[30] Damian, D., Zowghi, D., Vaidyanathasamy, L. and Pal, Y. "An industrial case study of immediate benefits of requirements engineering process improvement at the Australian Center for Unisys Software". *Empirical Software Engineering. vol.* 9(1-2), pp. 45-75, 2004.

[31] Niazi, M. and Babar, M.A. De-motivators of software process improvement: An analysis *Product* of Vietnamese-*Focused Software Process Improvement.* Springer, 2007.

[32] Hickey, A.M. and Davis, A.M. "Requirements elicitation and elicitation technique selection: model for two knowledge-intensive software development processes". *International Conference on System Sciences 2003.* Hawaii. *2003.*

[33] Sim, S.E., Alspaugh, T.A. and Al-Ani, B. "Marginal notes on a methodical requirements engineering: what experts learned from

experience". *IEEE International Requirements Engineering Conference*, 2008.

[34] Liu, L., Li T. and Peng, F. "Why requirements engineering fails: A survey report from china". *IEEE International Requirements Engineering Conference 2010. 2010*.

[35] Rahman, A.A., Haron, A., Sahibuddin, S. and Mazlan, M.H. "An Empirical Study of the Software Project Requirements Engineering Practice in Malaysia Public Sector–A Perspective from the International Stakeholders". *Journal of Computer Theory and Engineering. vol.* 6(1), 2014.

[36] Khankaew, S. and Riddle, S. "A review of practice and problems in requirements engineering in small and medium software enterprises in Thailand. *IEEE 4th International Workshop on Empirical Requirement Engineering (EmpiRE) 2014*. 2014.

[37] Zainol, A. and Mansoor, S. "Investigation into requirements management practices in the Malaysian software industry". *International Conference on Computer Science and Software Engineering 2008*, 2008.

[38] Hussain, A., Mkpojiogu, E.O.C. and Kamal, F.M. "Eliciting user satisfying requirements for an e-health awareness system using kano model". *International Conference on Computer and Computational Science (WSEAS) 2015*. Kuala Lumpur. pp. 156-165, 2015.

[39] Mkpojiogu, E.O.C. and Hashim, N.L. "Understanding the relationship between Kano model's customer satisfaction scores and self-stated requirements importance". *SpringerPlus*. Vol. 5(1), pp. 1-22, 2016.

[40] Mkpojiogu, E.O.C. and Hashim, N.L. "Quality-based prioritization: An approach for prioritizing software requirements". *Advancement on Information Technology International Conference* (*ADVCIT'15) 2015*. Krabi, Thailand. 3-5 December 2015.

[41] Hussain, A. and Mkpojiogu, E.O.C. "An application of Kano method in the elicitation of stakeholder satisfying requirements for an e-Ebola awareness system". *International Journal of Systems Applications, Engineering and Development*. Vol. 10, pp. 169-178, 2016.

[42] Hussain, A. and Mkpojiogu, E.O.C. "Requirements model for an e-health awareness portal". *International Soft Science Conference (ISSC) 2016.* Langkawi, Malaysia, 11-13 April 2016.

[43] Hussain, A. and Mkpojiogu, E.O.C. "Predicting the perceived worth of software products requirements with customer satisfaction". *Advanced Research in Engineering and Information Technology International Conference (AREITIC'16)*. Bandung, Indonesia. 31 May- 2 June 2016.

[44] Hussain, A., Mkpojiogu, E.O.C. and Hassan, F. "Assessing the influence of self-reported requirements importance on the perceived quality of proposed software products". *International Conference on Information and Communication Technology for Transformation (IC-ICT4T'16)*. Sabah, Malaysia. 5-7 April 2016.

[45] Hussain, A., Mkpojiogu, E.O.C. and Husin, Z. "Requirements: Towards an understanding on why software projects fail", *1st Int'l Soft Science Conference (ISSC'16)*, Langkawi Island, Malaysia, 11-13 April 2016.

[46] Hussain, A., and Mkpojiogu, E.O.C. "The effects of proposed software product's features on the satisfaction and dissatisfaction of potential customers". *International Conference on Applied Science and Technology (ICAST'16).* Sintok, Malaysia. 17 May 2016.