Structuring Challenges in Requirement Engineering Techniques

Olatunji J. Okesola, Kennedy O. Okokpujie, Paul R. Oyom, Olamide Kalesanwo, Oludele Awodele, Afolashade Kuyoro

***Abstract*— Experts in the field of requirement engineering have called for advancement in requirement structuring techniques as majority of the techniques are inadequate and give inconsistent results. This paper reviews the advancement in these techniques by answering the following questions: What is requirement structuring? What are the benefits of requirements structuring in the software engineering process? Who are the stakeholders involved the requirements structuring process? When do we structure requirements and what are the techniques and methods involved in requirements structuring? Addressing these questions from the requirement structuring perspectives, issues on software engineering were highlighted and discussed.**

***Index Terms*—requirement engineering, requirements structuring, software engineering, structuring techniques.**

I. Introduction

ALTHOUGH requirements aid the development of software, the general rule of thumb on how to develop software seems to be missing. According to [1] premature designs and solution based features are prominent. In software engineering, the objective of structuring requirements in any software development process is to develop requirements that fit the features or attributes of the problem itself [2]. He also stated that one interesting aspect of software engineering and software development is that the way requirements are structured and represented has an influence on the requirement activities.

One of the main causes of user discontent with delivered systems has been attributed to poor identification of requirements and unfinished requirements [3]. The requirement engineering process is vital in ensuring that there is a flow of communication among the stakeholders involved in the software development process and structuring requirements help in analysing, verifying as well as modifying those requirements [4].

The intention of this paper is to highlight the challenges in structuring requirements and it is structured as follows; the first section focuses on what structuring is in software engineering. The next section attempts to explain why requirement structuring is necessary. Next the paper looks at the actors or stakeholders involved in requirements structuring and also the roles they play in its implementation. The subsequent sections address when to structure requirements, the various instruments or techniques used in structuring requirements with a concluding statement.

II. What is Requirement Structuring?

The inability to identify requirements in advance leads to subsequent revaluation and rebuilding of the software [5]. In order to develop requirements that fits the system, one should have a proper guideline or process governing how such requirement are built.

Requirement structuring is a process of reducing the complexity of a system by having well classified and ordered requirements for any software development process. Requirement structuring ensures that the software development process is unambiguous, is consistent, complete and can easily be modified [7]. Requirements are often structured based on various criteria using hierarchies [6]. Requirement structuring help the stakeholders, analyse the impact of the project, merge the aim of the project to the solution, and also help the stakeholders reuse the knowledge gained from those requirements. Through requirements structuring, information is acquired from many sources as possible by the analyst. It also helps the analyst develop an initial scope of the project on time, enable large amounts of information gathered to be organized and also, alternatives to the development of the project can be identified.

III. Why do we Structure Requirements?

One bugging issue with customer dissatisfaction with software is that they fail to see the main purpose of a system. Most users are only concerned with the features of the proposed system and not what the system is expected to do. Structuring requirements help us understand the needs of the actors or stakeholders thereby putting more focus on the problem domain rather than the solution domain i.e. a solution based requirement [8]. A solution domain focuses on the features of the product. For instance, a user might request an extended feature for a mobile phone, such feature

Manuscript received December 27, 2017; revised January 10, 2018.

J.O. Okesola is with the Department of Computer and Information Sciences, Covenant University, Nigeria; e-mail: Olatunji.okesola@covenantuniversity.edu.ng.

K.O. Okokpuji is with the Department of Electrical and Information Engineering, Covenant University, Nigeria; e-mail: Kennedy.okokpujie@covenantuniversity.edu.ng

P.R. Oyom is with the Department of Computer and Information Sciences, Covenant University, Nigeria; e-mail: paulroyom@yahoo.comAyo.adebiyi@covenantuniversity.edu.ng

O. Kalesanwo is with the School of Computing and Engineering Sciences, Babcock University, Nigeria; e-mail: olamecome@gmail.com

O. Awodele is with the School of Computing and Engineering Sciences, Babcock University, Nigeria; e-mail: awodeleo@babcock.edu.ng

A. Kuyoro is with the School of Computing and Engineering Sciences, Babcock University, Nigeria; e-mail: kuyoros@babcock.edu.ng

is a distinct characteristic of that phone [1].

Requirements are structured to figure out the ways in which the user interacts with the system. According to [9], use cases show the interactions between a system and its external actors. Those interactions show possible scenarios and requirement structuring with use cases help to identify the possible scenarios that might arise from one interaction. Successful transitioning from the requirements to the finished products to ensure completeness and correctness is an important aspect in structuring requirements. [10]. With this in mind, the product complexity and the number of stakeholders involved are thoroughly considered. Once the reasons for structuring requirements are established, the next issue to be looked at is to identify the stakeholders involved in structuring requirements.

IV. WHO STRUCTURES REQUIREMENTS?

When structuring requirements, it is not possible to identify all stakeholders, they are later revealed during the course of the development process. [1]. Requirement structuring comprises of customers, users, domain experts, software engineers, project managers, developers, and requirement engineers as tabulated on Table1.

V.WHEN TO STRUCTURE REQUIREMENTS

To ensure all requirements are considered in the software development process when structuring requirements, it is important to note the following points.

- The requirements vary at each stage of the life cycle
- The requirements also change according to their level of specification.
- All requirements may not be identified by the stakeholders until the project is almost completed [9].

The requirements can be broken down into several categories to ensure each requirement match their level of specification. [10] developed a 10-dimensional model for structuring requirements. They argued that requirements cannot be acquired merely on only one dimension. The 10-dimensional model allows for easy structuring and determining of requirements thereby reducing the complexity of the product.

The model described in [10] was developed so the dimensions can be adapted to match requirements according to their level of specification. The dimensions, obligations and surroundings deal with requirements concerning with environmental aspects governed by law and those aspects not governed by law.

In “economy”, the items considered are the requirements concerning economic aspects; the cost-effectiveness and sustainability of a product. Under technical-functional aspects, requirements are categorised according to reliability, functionality and robustness. [12, 24, 29] defined reliability of a product as the ability of that product to satisfactorily perform its functions in a specified period of time.

“Product” includes requirements structured into the following categories: product core, formal product, extended product. For “weighted level of performance” the actual condition and satisfaction of the requirement is checked. [10, 23, 25, 26, 29] explained that the model surveys the stakeholder requirements and can also be used to identify requirements on particular topics from interviews with stakeholders.

Table 1: The roles and competences of stakeholders in requirement structuring

Stakeholder	**Intervention**	**Roles**	**Expertise**
Requirement analyst	Involved in the whole process	• Writes requirement specifications that are clear and unambiguous • Defines the constraints of the project • Identifies errors in the requirement document and report on them [16]	Analysis
Customer	Maintenance	Identify the requirement from multiple viewpoint [22]	Communication
Project Manager	Inspection	• Provide training for the stakeholders • Prioritize requirements according to level of importance [3]	Management, communication
Domain Expert	Validation	• Provide expert opinion on project • Identifies the functional requirement [17]	Decision making
End user		• Responsible for providing feedback [19]	
Developers		• Involve stakeholders throughout the development process [21].	Communication
Requirement manager		• Identify minimum needed requirement • Provide inspections for the requirement document [20 • Involves the users and customers throughout the development process [18]	Communication

VI. How to Structure Requirements

Requirement structuring is borne out of the need to guide the software development process. Requirement structuring can be undertaken when any organisation that has multiple projects ranging from maintenance projects to medium scale projects. Different requirement documents are scaled according to their complexity, risks as well as cost.

Structuring requirements have a number of approaches and have been used over time in maintaining consistency in projects. One of such methods is modelling. It is worth noting that models are not requirements but are used to complement efforts in structuring the requirements.

The earlier methods of requirement structuring include

- Process modelling with Data Flow diagrams
- Data modelling with Entity-Relationship diagrams
- Logic modelling with Structure English, decision trees, and decision tables

One widely adopted method for system analysis is the Rapid Application Development (RAD) method [13]. RAD speeds up the software development process by having the analysts and end users achieve common goals in due time. [14] observed that one advantage of the RAD tool is that the developers, designers and users can easily measure the progress gained on the project as a whole but a major setback is the difficulty experienced when used with large projects [14, 23].

Another method is the Object-Oriented Analysis (OOA) methods which specifically states what the system is expected to perform and not how it does it. The OOA approach aids in generating the requirements of the system thus identifies the functionality of the system. It brings everybody up to speed on the project with the simple design principles. The objective is to improve the quality of the software development process by capturing the structure of the processes involved. One disadvantage is that it can prove to be too complex for simpler projects because a project with simpler needs might not involve a structured approach [15].

VII. Summary of approach

Several literatures have highlighted varying approaches in gathering requirements. [2] stated that when structuring requirements, the set of requirements for that given project should be taken into account. Subsequently, the requirement architecture should be given the utmost consideration. This will go a long way in checking redundancy and reducing errors. [9] Opined that structuring use case with goals as a way to gathering and building requirements. The interactions between the various stakeholders and the system must be suited to a particular goal.

VIII. Conclusion

Over time, the focus on software engineering has been on how the system will achieve an objective rather what the system does under certain conditions. This has led to numerous software failures over the years.

This paper explores the concept of requirement structuring and highlights the challenges of encountered in software engineering. Also examined are the techniques in requirement structuring and the stakeholders involved and also various stakeholders in requirement structuring.

It is essential for all stakeholders involved to understand requirement structuring as an integral part in software engineering. The paper suggests that more object-oriented analysis approaches should be harnessed to shorten the software development process.

References

[1] J. Savolainen, T. Vehkomäki, and M. Mannion,: “An Integrated Model for Requirements Structuring and Architecture Design.” In: 7th Australian Workshop on Requirements Engineering, December 2-3, Deakin University, Melbourne (2002)

[2] S. Diev, “Structuring complex requirements”, (March, 2007). http://portal.acm.org/citation.cfm?id=1234741.1234755]

[3] D. Damian, “Challenges in requirements engineering”. Proceedings of 1995 IEEE International Symposium on Requirements Engineering (RE’95), 1997, pp. 160–162. https://doi.org/10.1109/ISRE.1995.512557

[4] N. Ramadan, and S. Megahed. “Requirements Engineering in Scrum Framework.” International Journal of Computer Applications, 2016, vol. 149, pp. 24–29. https://doi.org/10.5120/ijca2016911530

[5] S. Asghar, and M. Umar. “Requirement Engineering Challenges in Development of Software Applications and Selection of Customer-off-the-Shelf (COTS)”. Components. International Jounal of Software Engineering, vol. 1 ,pp. 32–50, 2010. http://www.cscjournals.org/csc/manuscript/Journals/IJSE/volume1/Issue2/IJSE-7.pdf?origin=publication_detail.

[6] E. Hull, K. Jackson and A. Dick. “Requirements Engineering. Requirements Engineering” Vol. 41. 2010, https://doi.org/10.1007/978-3-540-92201-8

[7] M. Geogy, and A. Dharani. “A Scrutiny of the Software Requirement Engineering Process.” Procedia Technology, vol. 25, pp. 405–410. 2016. https://doi.org/10.1016/j.protcy.2016.08.125

[8] S. Wiesner, S. Nilsson, and K. Thoben, Di. “Integrating Requirements Engineering for Different Domains in System Development - Lessons Learnt from Industrial SME Cases.” Procedia CIRP, vol. 64, pp. 351–356, 2017. https://doi.org/10.1016/j.procir.2017.03.013

[9] A. Cockburn, “Structuring use cases with goals. Journal of Object Oriented Programming”, vol. 5, pp. 1–16, 1997. http://alistair.cockburn.us/index.php/Structuring_use_cases_with_goals

[10] S. Klute, and R. Refflinghaus, “Structuring requirements as necessary premise for customer-oriented development of complex products: A generic approach”. Journal of Industrial Engineering and Management, vol. 4, pp. 523–537, 2011. https://doi.org/10.3926/jiem..v4n3.p523-537

[11] N. Gautam, and N. Singh, “Lean product development: Maximizing the customer perceived value through design change” (redesign). International Journal of Production Economics, vol. 114, pp. 313–332, 2006. https://doi.org/10.1016/j.ijpe.2006.12.070

[12] D. J. Smith, “Reliability, Maintainability and Risk”, pp. 29–37, 2011 https://doi.org/10.1016/B978-0-08-096902-2.00003-9

[13] B.Z. Zhu. “Requirements Determination and Requirements Structuring”. 2003.

[14] A. Powell-Morse. “Rapid Application Development (RAD): What is it and how do you use it?”, 2016, November 23. [Web log post] https://airbrake.io/blog/sdlc/rapid-application-development

[15] A. Powell-Morse. “Object-Oriented Analysis and Design: What is it and how do you use it?” 2017, April 18. [Web log post] https://airbrake.io/blog/design-patterns/object-oriented-analysis-and-design

[16] R. Young. “The Requirement Handbook”. Boston: Artech House. 2004.

[17] S. Maalem, and N. Zarour, “Challenge of validation in requirements engineering”. Journal of Innovation in Digital Ecosystems, vol. 3, pp. 15–21, 2016. https://doi.org/10.1016/j.jides.2016.05.001

[18] T. Huysegoms, M. Snoeck, G. Dedene, A. Goderis, and F. Stumpe. “Visualizing Variability Management in Requirements Engineering through Formal Concept Analysis”. Procedia Technology, vol. 9, pp. 189–199, 2013. https://doi.org/10.1016/j.protcy.2013.12.021

[19] J.O. Okesola, O.B. Longe and A.P. Obi. “Towards the Development of a Time-out Multiple C-R CAPTCHA Framework Using Integrated Mathematical Modelling”. *African Journal of Computing & ICTs*, vol. 8. iss. 2. pp 145-154, 2015. IEEE. http://www.ajocict.net/uploads/V8N2P17-2015_AJOCICT.pdf

[20] J.O. Okesola, O.S. Ogunseyeand O. Folorunso. “An Efficient Multi-Expert Knowledge Capture Technique”, *International Journal of Computer Applications (IJCA)*, vol. 8. iss. 10. pp 6-9. 2010. http://www.ijcaonline.org/volume8/number10/pxc3871611.pdf

[21] L. Ibidapo., A. Adebiyiand O. Okesola. Soft Computing Techniques for Stock Market Prediction: A Literature Survey”, Covenant Journal of Informatics and Communication Technology, Vol. 5 iss. 2. 2017. file:///C:/Users/Dr%20Tunji%20Okesola/Downloads/683-1199-1-SM.pdf

[22] A.A Owoade , O.F.W. Onifade , J.O. Okesola and B.L. Abimbola. “A Framework for Multimedia Data Hiding”. International Journal of Computer Science and Network Security (IJCSNS), vol. 11 iss. 12 pp. 99-104. 2011. http://paper.ijcsns.org/07_book/201112/20111215.pdf

[23] Osemwegie, O., Okokpujie, K., Nkordeh, N., John, S., Adewale Adeyinka, A. “On issues, strategies and solutions for computer security and disaster recovery in online start-ups” (2017) International Journal of Applied Engineering Research, 12 (19), pp. 8009-8015. https://www.scopus.com/inward/record.uri?eid=2-s2.0-85040244941&partnerID=40&md5=4075bca675bae7bb644c70976353066c.

[24] Osemwegie, O., John, S., Okokpujie, K., Shorinwa, I. “Development of an electronic fare collection system using stationary tap-out devices” (2017) Proceedings - 2016 International Conference on Computational Science and Computational Intelligence, CSCI 2016, art. no. 7881345, pp. 234-236. https://www.scopus.com/inward/record.uri?eid=2-s2.0-85017312311&doi=10.1109%2fCSCI.2016.0052&partnerID=40&md5=0defce44a8b3a5423ef72a032047c5c2.

[25] Okokpujie K, Noma-Osaghae E, John S, Ajulibe A. An Improved Iris Segmentation Technique Using Circular Hough Transform. In IT Convergence and Security 2017 2018 (pp. 203-211). Springer, Singapore.

[26] Okokpujie K., Etinosa NO., John S., Joy E. (2018) Comparative Analysis of Fingerprint Preprocessing Algorithms for Electronic Voting Processes. In: Kim K., Kim H., Baek N. (eds) IT Convergence and Security 2017. Lecture Notes in Electrical Engineering, vol 450. Springer, Singapore.

[27] O. Osemwegie, S. John, K. Okokpujie and I. Shorinwa, "Development of an Electronic Fare Collection System Using Stationary Tap-Out Devices," 2016 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, 2016, pp. 234-236.doi: 10.1109/CSCI.2016.0052.

[28] Chinonso, Okereke, Osemwegie Omoruyi, Kennedy Okokpujie, and Samuel John. "Development of an Encrypting System for an Image Viewer based on Hill Cipher Algorithm." COVENANT JOURNAL OF ENGINEERING TECHNOLOGY 1, no. 2 (2017).

[29] K. Okokpujie, E. Noma-Osaghae, S. John and R. Oputa, "Development of a facial recognition system with email identification message relay mechanism," 2017 International Conference on Computing Networking and Informatics (ICCNI), Lagos, 2017, pp. 1-6. doi: 10.1109/ICCNI.2017.8123776