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 requirements structuring in the software engineering process? Who are the stakeholders involved in 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 they seem 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].

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. 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
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 Table 1.

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.

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

<table>
<thead>
<tr>
<th>Stakeholder</th>
<th>Intervention</th>
<th>Roles</th>
<th>Expertise</th>
</tr>
</thead>
</table>
| 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]                             | Communication            |
| 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 born 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

http://portal.acm.org/citation.cfm?id=1234741.1234755


