Quality Assurance Techniques in SRS Documents

Prerana Chaithra, Shantharam Nayak

Abstract: ***For development of software, the most important aspects are the software requirements. They are the foundation stone for initiating any software development process. Software requirements documents contain the needs of the customers in natural language. By using various methods like reviews, inspections, walkthroughs, the content of the software requirement can be checked manually to reduce ambiguity. In recent years there is an attempt to automate these activities as a result of advancement in automation of natural language analysis. Automation of text mining techniques and text analysis is leading to feasibility of automation of requirements documents processing. The process can be completed in minutes now which were taking weeks earlier. Automation of analysis of text has triggered numerous possibilities for quality assurance of requirements. The possibilities of automation are model checking automation, automated rule checking, automated test case execution and measurement automation. In future more tools will enter the scene for automation of requirements quality assurance. At present most of them are in experimental stage. There is a definite need for more research in this field.***

Keywords: ***Ambiguity, Requirements document, Software Requirements, Quality assurance***

I. INTRODUCTION

Software requirements has major impact for determining the software quality. The errors in the requirement analysis phase of the software development, it easily gets propagated to other phases of the software. Therefore, removal of these ambiguities leads to better software. It makes the task of an analyst challenging to remove these ambiguities in the software requirements. It is always better to have less errors in the requirements to ensure less propagation of errors to make the task of the designer and coder easier. Among the documents required for the software development SRS is of greater significance. The remaining phases of the software development depends on SRS [1].

The ambiguous requirements lead to misunderstandings that leads to wrong misinterpretations implementations in design and development phase of the software [2].

If it is possible to implement such requirements, it is easy to know how much quality requirements are defined against functional requirements and their distribution across characteristic categories like compatibility, performance, usability, security and reliability. It is also possible to understand how structured they are in specification document [15].

Revised Manuscript Received on December 12, 2019.

Prerana Chaithra, Research Scholar, dept. of I.S.E, R.V.C.E., Bengaluru & Associate Prof., I.S.E., S.C.E. affiliated to VTU, Bengaluru, Karnataka, India. E-mail : preranachaithra15@gmail.com

Dr. Shantharam Nayak, Professor, dept. of I.S.E, R.V.C.E, Bengaluru, Karnataka, India. E-mail: shantaram_nayak@yahoo.com

The most important aspects for software development are the software requirements. They are the foundation stone for initiating any software development process. Software requirements documents contain the needs of the customers in natural language. The content of the software requirement can be checked manually by using various methods like reviews, inspections and walkthroughs. In recent years there is an attempt to automate these activities as a result of advancement in automation of natural language analysis. Automation of text mining techniques and text analysis is leading to feasibility of automation of requirements documents processing. The process can be completed in minutes now which were taking weeks earlier. Automation of analysis of text has triggered numerous possibilities for quality assurance of requirements. The possibilities of automation are model checking automation, automated rule checking, automated test case execution and measurement automation. In future more tools will enter the scene for automation of reduction of ambiguities in requirements.

In a software project development, requirements engineering is significant and effortful task. software requirements are generally mentioned using natural language(NL). There is a possibility that NL requirements may contain ambiguous and incomplete details [3]. To document software requirements, SRS is used that contain both non-functional and functional requirements. This document provides basis for next stages of software development process [16]. The defects are to be removed in early stage of development; otherwise the cost of fixing the errors in later stages would increase intensely.

For classification of requirements defects [21], it is necessary to have an adequate taxonomy to support the following tasks.

a) Identification of more frequent defects or the defects that have higher costs.
b) Analysis of the basic reason for the errors in the requirements.
c) Preparation of reviews of the requirements.

The quality parameters for requirements analysis are feasibility, comprehensibility, good structuring, modifiability, traceability, efficiency, correctness and completeness [10]. The Quality parameters and description are listed in the Table1.

Table1: Quality parameters

Quality Parameters	**Description**
Feasibility	In perspective of the financial plan, innovations and time frame, the requirements must be feasible.

Comprehensibility	The requirements must be understandable to by the general people who need to use them
Good Structuring	Link between components must be clearly specified
Modifiability	Should be convertible and adjustable
Efficiency	The requirements must fulfill the functionality and help in increasing the product performance
Correctness	The requirements which are to be developed must be correct and related to the functionality of the product.
Completeness	The requirements must be complete

Most of the existing techniques used for requirements engineering employ natural language process are concerned with SRS evaluation and improvement. NLP can be applied to detect ambiguity in words and sentences or phrases. NLP can also be used to use cases for ambiguity detection.

II. EXISTING TECHNIQUES

Assimilating the software requirements is the main task in software product development. Business analysts must develop the skill sets needed for progressing in their careers, for this they need to manage the ambiguities in the software requirements [12].

Ashima Rani et al [1] have developed a prototype for automatically detecting different kinds of ambiguities. The algorithm deals with Detecting different kinds of ambiguities automatically. It also finds the number of ambiguities of different types. It helps the business analysts to find different types of ambiguities in the SRS.

M. Q. Riaz, et al, [2] describes that Pre-processing phase which consists of identifying and detecting the ambiguous requirements in the SRS. It helps to remove the ambiguous requirements to have better requirements. Compound sentences and phrases makes natural language requirements very complicated.

Mohd Hafeez Osman, et. al. [3] describes method that is automated way of finding software requirements which have ambiguity. To remove this ambiguity, they have blended text mining and machine learning methodology. Text mining is used to train the dataset for extraction of the features. Using the training-set, this approach detects the ambiguity in the SRS. They have developed a working prototype for finding the ambiguous requirements. Fig. 1 depicts an approach for detecting ambiguous requirements.

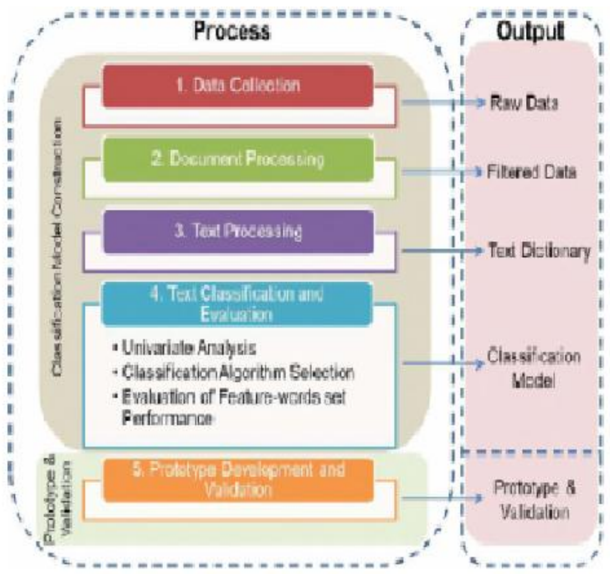

Fig. 1: Architecture for detecting ambiguous requirements

The first validation shows that the obtained result using the model for classification is moderately acceptable. This approach is quite acceptable as an experimental methodology.

Ali Olow Jim'ale Sabriye et al [4] clarify that if requirements can be interpreted in many ways then they lead to ambiguity in the software requirements. This leads to confusing state to developers and there is a possibility of developers building software that is different from what customer wants. In this work, there is a proposal for finding syntax as well as semantic ambiguities in the SRS. To detect the ambiguities, they have used the interpretation of the different parts of sentence. A prototype has been developed for evaluation of the proposed approach. The comparison has been done by comparing the capabilities of detection against human capabilities. The outcome of the result indicates that the humans do have some difficulties in ambiguity detection specifically syntactic ambiguity. The developed tool can be useful for analysts in detecting ambiguity in SRS documents.

T. Hovorushchenko et al [5] have explained the need of methodology development for sufficiency evaluation of information on SRS quality. The methodology developed evaluates how sufficient the SRS quality information is. It also asks for more information to increase the sufficiency of quantity of information. Result shows the need of methodology development for sufficiency evaluation of information on SRS quality.

R. Navarro-Almanza et al [7] explain that the quality of requirements decides the quality of the software being developed as the SRs represent the expectations and needs in a detail. Deep Learning(DL) has been used for classifying the requirements. They proposed an approach keeping Convolutional Neural Network(CNN) as the foundation. The results are partially good on SRC.

Azlin Nordin et al [8] did the analysis of quality properties of SRS. These factors are detected such that they are suitable for automation. They have detected requirements quality based on sentence and document. Relevant quality indicators were identified for each of the type. Later, further investigation is about the identification of documenting the indicators of quality.

A. Takoshima et al [10] say that the diversity in SRS leads to inspection that must be done systematically. They have proposed inspection in two phases. First phase of inspection is done by external inspectors. Members of the project team does the second phase of inspection. Due to this approach a different formats of software requirements can be inspected in the same way. The proposed approach has advantages like

1) SRSs are inspected from various product domains in a systematic way
2) Result analysis from several perspectives
3) Reference for SRS quality and results of inspection.
The researchers have evaluated the effectiveness of the method by applying them on two actual products.

Shalini Ghosh et al [11] developed an approach that generates formal model from natural language requirements. Benefits of the proposed model include redundancy/consistency checking and requirements validation, semantic ambiguities resolution and co-references in requirements, generation of examples for test cases.

Yue Wang et al [13] opine that identification of ambiguous requirements is a significant step in software development process. Errors in the later phases of software development which are costly to rectify are prevented. There are just handful of methods for solving the problem of identifying ambiguous requirements. In this work, lexical ambiguity detection problem is studied, and a method for detecting ambiguity in SRS using automation is proposed. Extensive focus is on ambiguities of lexical types.

III. RESULT AND DISCUSSION

Table 2 depicts the comparison of existing methods for quality in Software requirements.

Table 2: Comparison of existing methods

Author	Highlight of the work	Limitations
Ashima Rani et al 2019 [1]	• They have developed a prototype for automatically detecting different kinds of ambiguities. • The algorithm deals with Detecting different kinds of ambiguities automatically. • It also finds the number of ambiguities of different types. • It helps the business analysts to find different types of ambiguities in the SRS.	It is just an algorithm and is not yet implemented
M. Q. Riaz, et al 2019 [2]	• Pre-processing phase which consists of identifying and detecting the ambiguous requirements in the SRS. • It helps to remove the ambiguous requirements to have better requirements.	Sometimes difficult phrases and Compound sentences makes natural language requirements very complicated.
Mohd Hafeez Osman, et. Al. 2018 [3]	• They have described method that is automated way of finding software requirements which have ambiguity. • To remove this ambiguity, they have blended text mining and machine learning methodology. • Text mining is used to train the dataset for extraction of the features.	This study only focuses on the Malaysian context. It is not flexible.
Ali Olow Jim’ale Sabriye et al 2018 [4]	• There is a proposal for a methodology for finding syntax ambiguities as well as semantic ambiguities in the SRS. • To detect the ambiguities, they have used the interpretation of the different parts of sentence. • A prototype has been developed for evaluation of the proposed approach. The comparison has been done by comparing the capabilities of detection against	Results show that it is difficult to detect syntactic ambiguity and semantic ambiguity.

	human capabilities.	
T. Hovorushchenko et al 2018 [5]	• They have explained the need of methodology development for sufficiency evaluation of information on SRS quality. • The methodology developed evaluates how sufficient the SRS quality information is. • It also asks for more information to increase the sufficiency of quantity of information. Result shows the need of methodology development for sufficiency evaluation of information on SRS quality. •	Result shows that there is a need for developing a method for sufficient information to be evaluated to determine SRS quality.
R. Navarro-Almanza et al 2017 [7]	• Used Deep-Learning has been used for classifying requirements. • They proposed an approach keeping Convolutional Neural Network(CNN) as the foundation	The results are partially good on SRC
Azlin Nordin et al 2017 [8]	• They have the analysis of quality properties of SRS. • These factors are detected such that they are suitable for automation. • They have detected requirements quality based on sentence and document. Relevant quality indicators were identified for each of the type. • Further investigation is about the identification of documenting the indicators of quality.	Results show the identification of relevant quality indicators that are requirements that are feasible but not yet automated.
A. Takoshima et al 2015 [10]	• They have proposed inspection in two phases. First phase of inspection is done by external inspectors. • Members of the project team does the second phase of inspection. • Due to this approach a different formats of software requirements can be inspected in the same way.	Results show that SRSs are inspected from various product domains in a systematic way, but they are still manual and not automated. Result analysis from several viewpoints Benchmark of SRS quality and inspection results.
Shalini Ghosh et al 2014 [11]	• They have developed an approach that generates formal model from natural language requirements. • Benefits of the proposed model include redundancy/consistency checking and requirements validation, semantic ambiguities resolution.	Results show that they are only partially automated and does not always work correctly
Yue Wang et al 2013 [13]	• Proposed methods to identify potentially ambiguous concepts in SRSs. • In this work, lexical ambiguity detection problem is studied, and a method for detecting ambiguity in SRS using automation is proposed.	Extensive focus is on ambiguities of lexical types.

It can be observed from the previous studies that there exist mechanisms to automate the quality measurement of software requirement specification. Previous studies [10] have proposed methods to identify potentially ambiguous concepts in SRSs. In one of the works [8], the automatic detection is done using txt format.

There is a need for development of methods that can accept other formats such as doc and pdf formats. There is a need to take into account of other ambiguity like scope ambiguity and lexical ambiguity.

IV. CONCLUSION

Efforts to control the requirement document can be seen from long time. The researchers demonstrated that earlier detection of defects saves costs and enables the developers to easily remove them. In recent days, several works can be seen to automate the analysis of requirements documents. Several papers have been published in the area of automatic quality detection of software requirement documents. There are tools developed to recognize contradictions and missing attributes. Consistency and completeness check as well as preciseness tests needs to be automated. In this work, an effort has been made to walk through the existing techniques in automation of quality checking of software requirements documents. The highlights of the existing techniques have been discussed and research gaps have been identified. It can be concluded that there exists a lot of scope for improvements in automating the quality checking of the software requirement specification. It can also be noted that there are some quality attributes that are not feasible to be automated. In such instances, human intervention is required.

REFERENCES

1. Ashima Rani, Gaurav Aggarwal “Algorithm for Automatic Detection of Ambiguities from Software Requirements”, IJITEE, ISSN: 2278-3075, Volume-8, Issue-9S, July 2019, pp. 878-882.
2. M. Q. Riaz, W. H. Butt and S. Rehman, "Automatic Detection of Ambiguous Software Requirements: An Insight," 2019 5th International Conference on Information Management (ICIM), Ambridge, United Kingdom, 2019, pp. 1-6.
3. Mohd Hafeez Osman and Mohd Firdaus Zaharin, “Ambiguous Software Requirement Specification Detection: An Automated Approach”, IWRET, ACM, New York, USA, 2018, pp. 33-40.
4. Ali Olow Jim’ale Sabriye, Wan Mohd Nazmee Wan Zainon, “An approach for detecting syntax and syntactic ambiguity in software requirement specification”, Journal of Theoretical and Applied Information Technology, Vol.96. No 8, 2018, pp 2275-2284.
5. T. Hovorushchenko and O. Pomorova, “Methodology of evaluating the sufficiency of information on quality in the software requirements specifications”, IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT), Kiev, 2018, pp 370-374.
6. Tetsuo Tamai1 and Taichi Anzai, “Quality Requirements Analysis with Machine Learning”, in the Proceedings of the 13th International Conference on Evaluation of Novel Approaches to Software Engineering, 2018, pp 241-248.
7. R. Navarro-Almanza, R. Jurez-Ramrez and G. Licea, “Towards Supporting Software Engineering Using Deep Learning: A Case of Software Requirements Classification”, 2017 5th International Conference in Software Engineering Research and Innovation (CONISOFT), Mérida, 2017, pp 116-120.
8. Azlin Nordin, Nurul Husna Ahmad Zaidi and Noor Asheera Mazlan, “Measuring Software Requirements Specification Quality”, Journal of Telecommunication, Electronic and Computer Engineering, Vol.9, No.3-5, 2017, pp 123-128.
9. Uzma Noorin & Mehreen Sirshar, “Quality Assurance in Requirement Engineering”, Global Journal of Computer Science and Technology: C Software & Data Engineering, Volume 17 Issue 1 Version 1.0, 2017, pp.1-6.
10. A. Takoshima and M. Aoyama, “Assessing the Quality of Software Requirements Specifications for Automotive Software Systems”, 2015 Asia-Pacific Software Engineering Conference (APSEC), New Delhi, 2015, pp 393-400.
11. Shalini Ghosh, Daniel Elenius, Wenchao Li, Patrick Lincoln, Natarajan Shankar, Wilfried Steiner, “Automatically Extracting Requirements Specifications from Natural Language”. 2014. https://arxiv.org/abs/1403.3142v1
12. P. Bourque and R. E. Fairley, “Guide to the Software Engineering Body of Knowledge (SWEBOK (R)): Version 3.0”. IEEE Computer Society Press, 2014. pp 27- 40.
13. Yue Wang, Irene L. Manotas Gutierrez, Kristina Winbladh, and Hui Fang, “Automatic Detection of Ambiguous Terminology for Software Requirements”, International Conference on Application of Natural Language to Information Systems, 2013, pp 25-27.
14. Bechoo Lal, Dr. Chandrahauns R. Chavan, “An Optimization Approach to Analysis of Requirement”, IJARCSSE International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 2, February 2013, pp 311-316.
15. Fatwanto A., “Software requirements specification analysis using natural language processing technique”, International Conference on QiR (Quality in Research) at Yogyakarta on 25-28 June 2013, IEEE, print ISBN 978-1-4673-5784-5, pp. 105 – 110.
16. Misra J., Das S., “Entity Disambiguation in Natural Language Text Requirements”, 20th Asia-Pacific Software Engineering Conference, IEEE, Volume 1, 2013, pp. 239 – 246.
17. Ayan Nigam, Neeraj Arya, Bhawna Nigam and Deepika Jain, “Tool for Automatic Discovery of Ambiguity in Requirements”, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 2, September 2012, pp. 350-356.
18. Harter D.E., Kemerer C.F., Slaughter S.A., “Does Software Process Improvement Reduce the Severity of Defects? A Longitudinal Field Study”, Transactions on Software Engineering, IEEE, Volume 38, Issue 4, July-Aug. 2012, pp. 810 – 827.
19. Isabel Lopes Margarido, Marco Vieira, “Classification of Defect Types in Requirements Specifications: Literature Review, Proposal and Assessment”, Vol. I, CISTI 2011 - 6ª Portugal; 2011, pp. 555-561.
20. Adriana Beal, “Managing ambiguity – a key business analyst competency”, Tips for growing your Business Analyst Career, Internet Draft, 2010. http://www.bridging-the-gap.com/managing-ambiguity-a-key-business-analyst-competency
21. S Park, HKim, YKo, J Seo, “Implementation of an efficient requirements-analysis supporting system using similarity measure techniques”, Information and Software Technology, ELSEVIER, Volume 42, Issue 6, 15 April 2000, pp 429–438.

AUTHORS PROFILE

Prof. Prerana Chaithra is a Research Scholar at R.V.C.E. Bengaluru & Associate Prof. in the department of I.S.E. at S.C.E., affiliated to VTU, Bengaluru. She has done her B.E., M. Tech and pursuing her PhD in CSE. She has 3 years of Industrial Experience and 19 years of Academic experience both in UG and PG. She worked as Head of the Department for 10 years. She has won 2 International awards in the field of Engineering and 2 National awards for her contribution to the society. To her credit she has 25 Journal as well as conference papers in International level and National level. 5 of her papers have received first and second prizes. She is pursuing her research work in SoftwareEngineering and Data Analytics in C.S.E. branch at R.V.C.E. under the Guidance of Dr. Shantharam Nayak, who is a pioneer in the field of SoftwareEngineering.

Dr. Shantharam Nayak, Professor, I.S.E., R.V.C.E, Bengaluru has done his BE., MS. and PhD in CSE. He has 27 years of Academic experience both in PG & UG. He is guiding PhD students and some have already completed their PhD. To his credit he has 38 Journal papers in International level, 15 conference papers both in International and National level. He is a knowledgeable professor in the field of Software Engineering. He is associated with CSI since 1992. He has organized more than 90 invited talks, has coordinated & conducted 2 International & 6 National conferences, 11 FDP’s, 18 workshops at R.V.C.E. He was responsible for starting C.S.I. student-branch in R.V.C.E. and bagged “C.S.I. Best student-branch Region-5” award continuously for eight times since 2005-06. He is honored with Fellow of IE; Significant contribution, Longest Continuous SBC & Patron Award from CSI.