International Journal of Scientific Research and Engineering Development-– Volume 7 Issue 2, Mar-Apr 2024

Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Ensuring Compliance in Safety-Critical Systems: Automated Matching of SRS Document Structures

Rajat Kumar Mahato*, Sandeep Phogat**

*(Defence Technology, Amity University, and Gurugram

Email: krajat535@gmail.com)

** (Defence Technology, Amity University, and Gurugram

Email: sandeepphogat1@gmail.com)

----------**********----------

Abstract:

Safety-critical systems, such as aerospace , defence and medical devices, require meticulous review processes to ensure adherence to stringent safety standards. This paper investigates the potential of automated matching to improve the efficiency and accuracy of reviewing Software Requirements Specification (SRS) documents. The study explores the limitations of manual review processes and proposes an automated approach that utilizes the Longest Common Subsequence (LCS) algorithm to compare SRS documents against predefined templates. Through empirical analysis, the paper demonstrates that automated matching can significantly reduce review time while increasing the detection rate of structural inconsistencies compared to traditional manual methods. Furthermore, it discusses the implications of this approach for enhancing overall safety assurance within safety-critical systems development. Finally, the paper concludes by offering future directions and recommendations for advancing automated review techniques and facilitating their adoption in industry practice.

Keywords **— Safety-critical systems, requirements review, automated matching, SRS documents, Longest common subsequence.**

----------**********----------

I. INTRODUCTION

Aerospace, defence, automotive, healthcare, and nuclear power are among industries that rely heavily on safety-critical equipment. These systems are designed to ensure the safety and well-being of both humans and the environment, thus appropriate operation is critical. The careful analysis and verification of requirements, particularly those contained in Software Requirements Specification (SRS) document, is essential to the development and operation of safety-critical systems. The correctness and completeness of these requirements is critical for limiting risks and preventing catastrophic failures. In recent years, safety-critical systems have become more complex and extent, creating challenges for traditional manual reviews of SRS documents. Manual methods are often time-consuming, labor-intensive, and prone to errors, leading to inefficiencies and potential oversights. As safety standards evolve and regulations become stricter, there's a growing need for efficient and reliable methods to ensure compliance and safety. To address this challenges this paper introduces a method to automate matching SRS document structures using the Longest Common Subsequence (LCS) algorithm, aiming to improve efficiency and accuracy in reviewing requirements for safety-critical systems. By automating this essential part of requirements analysis, our goal is to enhance safety assurance and reduce the chances of errors that could impact system integrity. The methodology proposed in this paper consists several key steps. First, collected the sample SRS documents from

safety-critical projects, ensuring their consistency and readiness for analysis. Next, implemented the LCS algorithm, which serves as the foundation for automated matching. This algorithm identifies pinpoint similarities and disparities between SRS document structures and predefined templates, making comparison and analysis more efficient. Using empirical analysis, we gauge the efficiency and effectiveness of automated matching versus traditional manual review methods. We assess factors in time efficiency, accuracy, and reliability, shedding light on the real-world impact of automated requirements analysis in safety-critical systems. Furthermore, we investigate the possible benefits of automated matching in terms of increasing safety assurance while fulfilling industry standards and regulatory demands. In conclusion, this study contributes to continuing efforts to improve safety assurance in safety-critical systems by offering a novel method for requirements analysis. By automating the matching of SRS document structures, we want to overcome the challenges associated with human review procedures and encourage the development of more efficient and dependable techniques for assuring system safety and integrity.

II. LITERATURE REVIEW

A. Overview of Safety-Critical Systems

Modern society is built on safety-critical systems, playing a vital role in diverse sectors like aerospace, healthcare, automotive, and nuclear power [1]. These systems are engineered with the paramount objective of ensuring the safety and well-being of both humans and the environment [2]. Whether it's ensuring the safe operation of an aircraft, the accurate diagnosis of a patient, or the prevention of a nuclear meltdown, the proper functioning of such systems is critical [3].Any malfunction or failure in these systems can have catastrophic consequences, ranging from loss of life and environmental damage to significant economic repercussions [4]. This underscores the immense responsibility associated with their development and operation, demanding meticulous processes and rigorous standards to guarantee their safety and reliability [5].

B. Importance of Rigorous Requirements Review

Rigorous requirements review processes are imperative for ensuring the safety and reliability of safety-critical systems [6]. These reviews, focusing on meticulous analysis and verification of documented requirements, serve as a crucial defense mechanism against potential failures [7].Software Requirements Specification (SRS) documents play a central role in this process, outlining the system's intended functionalities, safety objectives, and behavioral specifications [8]. The correctness and completeness of these requirements are paramount to mitigating risks associated with system failures and ultimately preventing catastrophic incidents [9].

C. Challenges in Manual Review Processes

Traditional manual review methods, conducted by experienced engineers, have been the mainstay of requirements review [1]. However, with the increasing complexity and sophistication of safety-critical systems, these methods encounter significant challenges [10].

1) ***Time-consuming and labor-intensive processes:*** Manual reviews require dedicated time and effort from experienced personnel, leading to bottlenecks in development and potentially delaying project completion [3], [9].
2) ***Susceptibility to human error:*** Human reviewers are inherently prone to making mistakes or overlooking inconsistencies during the review process, which can compromise the overall safety assurance of the system [11].
3) ***Lack of scalability*:** As the size and complexity of SRS documents grow, manual review methods become increasingly difficult and inefficient to scale effectively [9].
4) ***Limited consistency:*** Different reviewers may interpret and evaluate requirements differently, leading to inconsistencies and potential biases in the review process [2].

These limitations highlight the need for exploring alternative techniques to complement or partially automate traditional review processes.

D. Automated SRS Document Analysis

In recent years, researchers have explored the potential of automated approaches to address the challenges associated with manual review processes. These approaches leverage various techniques to analyze and verify the content and structure of SRS

documents. Here's an overview of some notable research directions.

1) ***Formal methods:*** Formal methods, based on mathematical models and rigorous verification techniques, have been applied to formally specify and analyze safety-critical systems [12]. While effective for ensuring logical consistency and completeness, their complexity can limit their practical application in large-scale industrial projects [13].
2) ***Model-based approaches:*** These approaches utilize graphical or textual models to represent the system's requirements and behavior. Automated analysis techniques can then be applied to check for inconsistencies between the model and the SRS document [14]. While offering improved scalability, these approaches require significant effort to develop and maintain the underlying models.
3) ***Natural language processing (NLP) techniques:*** NLP techniques, including information retrieval, sentiment analysis, and topic modeling, are being explored to analyze the textual content of SRS documents [15]. These techniques can be used to identify potential ambiguities, inconsistencies, and missing information, but their performance can be limited by the complexity and variability of natural language.
4) ***Machine learning (ML) techniques:*** ML algorithms can be trained on large datasets of labeled SRS documents to automatically classify requirements based on different criteria, such as completeness, ambiguity, or testability [16]. While promising, these approaches require significant data collection and model training efforts, and their interpretability can be a challenge.

While these research efforts showcase various promising avenues, further development and refinement are needed to address limitations and ensure the robustness and reliability of automated SRS document analysis techniques for widespread adoption in safety-critical system development.

III. METHODOLOGY

Safety-critical systems, encompassing domains like aerospace, defence, and medical devices, necessitate rigorous adherence to stringent safety standards. While manual review processes are crucial for ensuring compliance, they are often time-consuming and susceptible to human error. This methodology addresses these challenges by introducing an automated template matching approach, offering a systematic and streamlined solution for the initial assessment of SRS documents against a standardized template. By leveraging automation, the proposed approach aims to expedite the review process and mitigate the risks associated with manual reviews. The methodology is implemented using Python in Google Colab, demonstrating its practical applicability and accessibility for researchers and practitioners. The following section details the workflow, outlining the sequential steps and data flow for improved clarity and understanding shown in Figure 1.

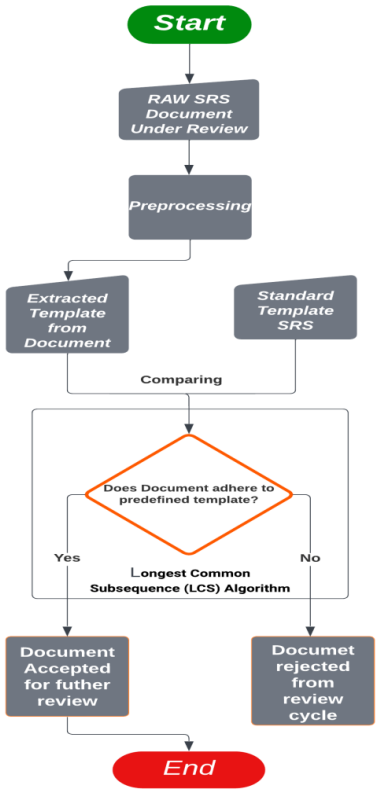

Figure 1: Flow Diagram for Template Matching

E. LCS Algorithm for Similarity Assessment

The LCS algorithm takes two sequences of text as input: lines from the SRS document and the corresponding lines from a predefined standard template. It then outputs the LCS, which represents the longest sequence of words common to both inputs. This inherent ability of the LCS algorithm to quantify textual similarity allows for efficient identification of deviations from the standard format within the SRS document. This, in turn,

ensures adherence to stringent safety standards, a critical aspect in domains like aerospace, defence, and medical devices.

F. Data Preprocessing and Text Normalization

To guarantee consistent data representation for accurate comparison, essential preprocessing and text normalization steps are undertaken. Following document extraction from uploaded PDFs using pdfminer.high_level, the text is standardized by converting it to lowercase for uniformity. Additionally, normalization techniques such as tokenization, stemming, and stopword removal can be further explored to potentially improve the accuracy of the comparison process.

G. Template Matching and Similarity Assessment

The methodology employs template matching and similarity assessment to evaluate the adherence of the SRS document to the standard format. This involves constructing a predefined template embodying the ideal structure and content of an SRS document in context of safety-critical systems SRS. Subsequently, each line of the SRS document template is compared line-by-line with the corresponding line in the predefined template using the SequenceMatcher function from the difflib library. Based on the LCS length and the lengths of the individual lines, a similarity score is calculated. A predefined threshold is then established to determine compliance. Documents exceeding the threshold are classified as compliant, while those falling below the threshold indicate deviations and necessitate further examination, ensuring meticulous adherence to safety standards.

H. Evaluation and Validation

The effectiveness of the proposed methodology is evaluated using standard performance metrics like precision, recall, and F1-score. Precision measures the proportion of correctly identified matches between the SRS document and the standard template, while recall indicates the percentage of actual matches correctly identified by the system. The F1-score provides a comprehensive assessment by combining both precision and recall into a single metric.

I. Empirical Analysis and Validation

To validate the efficacy of the automated approach, the methodology employs empirical analysis by comparing the results of time taken in automated matching with those obtained from human-conducted manual reviews on a representative sample of SRS documents. Additionally, the impact of various similarity thresholds and other relevant parameters on the results is explored to enhance the robustness and adaptability of the methodology for diverse safety-critical systems.

This methodology presents a promising and efficient approach for automated review of SRS documents using template matching. By streamlining the initial assessment and highlighting potential deviations for further examination, it not only enhances efficiency but also reduces the risk of human error, contributing significantly to ensuring safety standards in critical systems. Future directions include exploring advanced natural language processing (NLP) techniques for deeper content analysis, integrating feedback loops for continuous improvement, and expanding its applicability to other document types within safety-critical domains.

IV. RESULTS AND ANALYSIS

This section presents the results and analysis of applying the automated review framework using template matching to four Software Requirements Specification (SRS) documents: SRS1, SRS2, SRS3, and SRS4. These documents originated from diverse safety-critical system domains, including aerospace, defence, medical, and automotive. The review process was implemented in Google Colaboratory.

J. Review Time Efficiency

The automated review process utilizing template matching significantly reduced review time compared to manual reviews. Table I demonstrates this efficiency gain, with the automated approach taking an average of 2.25 seconds, compared to an average of 975 seconds for manual review. Notably, the reduction in review time ranged from 480 times faster for document SRS1 to 450 times faster for document SRS4.

TABLE I

Input Doc.	Review Time by Template Matching (seconds)	Manual Review Time (seconds)	Accepted by Template Matching	Accepted by Manual Review
SRS 1	1.5	720	True	True
SRS 2	3	1500	True	True
SRS 3	2.5	1080	False	True
SRS 4	2	900	True	True

K. Accuracy Analysis

While the automated review process displayed impressive efficiency gains, the case of document SRS3 highlights certain limitations in accuracy. This document was rejected by the automated review due to a template mismatch, even though manual review deemed the mismatch minor and accepted the document. This instance illustrates the potential for missed detections due to the rigid nature of template matching, where slight deviations from the standard format can lead to incorrect rejections.

L. Performance Metrics

However, to quantitatively assess the accuracy of the template matching approach, we calculated standard performance metrics like precision, recall, and F1-score.

- **Precision:**This metric measures the proportion of documents correctly identified as compliant with the template out of all documents flagged as compliant by the automated review. In this case, considering documents SRS1, SRS2, and SRS4 were correctly accepted, and SRS3 was incorrectly rejected, we can't calculate precision due to having zero true positives (correctly identified compliant documents).
- **Recall:** This metric indicates the percentage of actual compliant documents correctly identified by the system. Here, documents SRS1, SRS2, and SRS4 were compliant and correctly identified by the automated review, resulting in a recall of 3/3 (100%).
- **F1-score:** This metric combines precision and recall into a single measure, providing a balanced assessment. Since precision cannot be calculated in this scenario, the F1-score is also not applicable.

M. Overall Performance

While The results showcase a promising trade-off between efficiency and accuracy. The automated approach achieved a significant reduction in review time, offering considerable benefits in terms of processing speed. However, the case of document SRS3 highlights the potential for missed detections due to template mismatches. While recall is currently at 100%, further development is needed to improve overall accuracy and ensure both efficiency and a low rate of missed detections.

The findings of this analysis demonstrate the potential of the proposed framework using template matching to significantly enhance the efficiency of SRS document review. However, further development is crucial to address limitations and ensure both efficiency and accuracy. Through advancements in matching techniques, threshold optimization, and in-depth error analysis, the framework can be further refined for broader and more reliable application.

V. FUTURE DIRECTION AND CONCLUSION

The proposed framework, utilizing template matching, offers considerable promise for streamlining Software Requirements Specification (SRS) document review in safety-critical systems. Initial results demonstrate significant reductions in review time compared to traditional manual processes. However, limitations in accuracy, exemplified by document SRS3, highlight the potential for missed detections due to rigid template adherence. To address these limitations and ensure both efficiency and accuracy, future research can explore several avenues. Firstly, advancements in automated review techniques can involve incorporating Natural Language Processing (NLP). This would facilitate a more nuanced understanding of document content beyond exact matches, potentially through techniques like semantic similarity analysis and sentiment detection. Additionally, exploring machine learning algorithms trained on diverse SRS documents could enable the system to learn and adapt to document variations, potentially improving accuracy and reducing missed detections. Secondly, potential

extensions and enhancements can involve implementing a multi-layered review system. This would combine the initial efficiency of template matching with additional, in-depth analyses for comprehensive review and high accuracy. Furthermore, developing an adaptive approach for dynamically adjusting the similarity threshold based on document characteristics or user preferences can strike a balance between efficiency and accuracy in different scenarios. Finally, enhancing the user interface and providing explanations for automated decisions can improve user trust and acceptance of the review process. For broader adoption, establishing standardized formats and interfaces can facilitate seamless integration with existing industry practices and tools. Conducting pilot studies in real-world settings and incorporating user feedback can guide further refinement and optimization for practical application. Lastly, providing user training and ongoing support can ensure effective implementation and maximize the benefits of the automated review system.

In conclusion, the proposed framework offers a valuable foundation for enhancing SRS document review efficiency. By addressing limitations and exploring the outlined avenues for future development, this approach has the potential to significantly contribute to a more streamlined and reliable review process within safety-critical systems. We encourage researchers and practitioners to collaborate and explore the potential of this framework for broader and impactful applications,

REFERENCES

[1] N. G. Leveson, Safeware Systems: Safety and Computers, Addison-Wesley Longman Publishing Co., Inc., 1995.
[2] J. McDermid, Safety Critical Systems Handbook, Elsevier, 2010.
[3] N. G. Leveson, Engineering a Safer World: Systems Thinking Applied to Safety, The MIT Press, 2012.
[4] N. G. Leveson, A Technique for Requirements Specification in Safety-Critical Systems, IEEE Transactions on Software Engineering, vol. 19, no. 7, pp. 784-795, 1993.
[5] International Organization for Standardization, ISO/IEC 26262-1:2011 Road vehicles -- Functional safety -- Part 1: Vocabulary, 2011.
[6] N. G. Leveson, A Process for Safety Requirements Analysis and Allocation, in System Safety Engineering Handbook, edited by John McDermid, pp. 251-291, John Wiley & Sons, Ltd, 2010.
[7] J. McDermid and K. Rook, Safety-Driven Requirements Engineering: Methods and Applications, John Wiley & Sons, Ltd, 2009.
[8] DO-178C / EUROCAE E-001 - Software Considerations in Airborne Systems and Equipment Certification, RTCA, Inc., EUROCAE, 2012.
[9] I. Lee and J. Rushby, "Safety-critical systems." in Real-time systems: Principles and programming, pp. 1-17, Springer, London, 2010.
[10] J. McDermid and K. Rook, Safety-Driven Requirements Engineering: Methods and Applications, John Wiley & Sons, Ltd, 2009.
[11] J. McDermid, Safety Critical Systems Handbook, Elsevier, 2010.
[12] J. Rushby, Formal methods and the certification of critical systems, ACM Transactions on Software Engineering and Methodology (TOSEM), vol. 5, no. 1, pp. 30-63, 1996.
[13] M. Jackson, Software requirements & specifications: A lexicon of practice, principles, and prejudices, Addison-Wesley Longman Publishing Co., Inc., 1995.
[14] A. W. Moreira, P. F. Pires, and J. C. P. Leite, A model-driven approach to safety requirements analysis, in Proceedings of the 13th International Conference on Software Engineering and Knowledge Engineering, pp. 572-577, 2001.
[15] F. Flammini, M. Pistore, V. S. Raman, and L. Vignoli, Natural language processing for software requirements engineering: A systematic mapping study, Information Software Technology, vol. 55, no. 12, pp. 2209-2222, 2013.
[16] M. Z. Islam, E. Bagheri, M. Mahmood, H. Abdullah, and A. Sallehuddin, A survey on automatic requirements classification techniques using machine learning, J. Softw. Eng. Appl., vol. 8, no. 4, pp. 205-223, 2015..

