Requirements Specification Automated Quality Analysis : Past, Present and Future

Dylan Porter, Joanna DeFranco
Penn State

Phil Laplante
IEEE Fellow

Abstract:

The challenge of developing a solid requirements specification with desired qualities is discussed. Some past and present requirements engineering quality measurement tools are briefly reviewed. Desired functionality for next generation requirements engineering quality measurement tools is discussed and one such tool in development is described.

Requirements Specification Desired Quality Attributes

A solid requirements specification is essential in insuring that the desired system meets its functional, schedule and cost objectives. As systems become more complex, however, the development of requirements specifications is correspondingly more challenging. The complexity and associated challenges require automated tools to help assist in the requirements engineering process. Today, there are a wide range of uses for automated tools within the requirements domain.

Some tools helped to manage the requirements and organize the document, other tools attempt to measure certain quality attributes of the resultant document. Other tools assist in a variety of requirements engineering tasks, such as detection, extraction, classification, modeling, tracing, and requirements searching [1]. Some tools analyze the quality of software requirements and requirement groups. The tools utilized for detection of quality violations are extensive and wide-ranging.

Requirements Specification Attributes

For almost 30 years, IEEE 830-1998 [2] and its successor IEEE 29148-2011 [3] have designated attributes for measuring the quality of both requirements and requirement groups. These quality attributes help to formulate a better set of requirements by providing guidelines for ideal requirements and requirement groups. For instance, the standard says that an individual requirement should be feasible. This goal means that a requirement should be practical to implement within any budgetary or system restrictions. Another quality attribute that the standard provides is completeness, which says that a requirement should adequately describe the entirety of the user need [3].

The focus of many requirements tools is to measure the quality attributes outlined or associated with the IEEE 29148 standard, or other, similar documents. Although older tools have generally helped facilitate a better requirement engineering process, they were imperfect. Generally, a wide variety of these tools measured requirements quality through general rules-of-thumb. As a result, the tools analyzed the software requirements on a morphological, lexical, and syntactical basis.

Some of these tools had pre-defined words or phrases that they would check for to analyze requirements quality defects. For instance, various words such as 'it', 'this', 'that', could be considered ambiguous. Additionally, there are also certain phrases, such as 'user friendly' and 'easy to use' that also create quality attribute violations [3]. Researchers have extended the definitions and types of quality attributes used during analysis. Unfortunately, existing research indicates that common quality indicator to attribute mappings may not be entirely useful or accurate.

A Brief History Lesson

In the late 1990s the US National Aeronautics and Space Administration (NASA) Software Assurance Technology Center (SATC) developed a tool to analyze a requirements document and produce a detailed quality report. The Automated Requirements Measurement (ARM) was a foundational tool that utilized indicators such as imperatives, continuances, and weak phrases to analyze the quality of the requirements document. The tool maps the textual statistics related to those indicators to quality attribute violations. It also produced a report that was based on statistical analysis of word frequencies at various structural levels of the document. The ARM tool was later enhanced to include additional functionality such as custom definitions of quality indicators inputs for document analysis. By 2011 work on the ARM tool was discontinued.
In 2012 Phil Laplante and Nathan Carlson reconstituted a version of ARM. The reconstructed tool behaved nearly exactly as the old tool (with some corrections for errors discovered during the reconstruction [4]). The reconstructed version of the tool is still used in requirements engineering courses and in industry and government projects around the world and is available at ARM Tool (laplante.io). The host owner is regularly asked about the status and future development for the ARM project.

There were several other similar tools developed around this time such as the Requirements Quality Tool (RQT), Quality Analyzer of Requirements Specification (QuARS), and the FAA Requirements Quality Tool (FRQT) (a variant of QRT) [5]. These tools are mostly extinct.

Some quality analysis tools are available today including the Requirements Quality Analyzer (RQA) and Systemized Requirements Engineering Environment (SREE), but these are largely academic implementations.

While these tools can measure many of the 29148 quality attributes through pre-defined words and phrases, the tools fall short when context is critical to measure quality attribute violations. For instance, heuristics generally cannot measure completeness quality attribute violations, as measuring the quality attribute requires an understanding of context and system restrictions.

AI Enabled Requirements Specification Quality Measurement

Recent developments in artificial intelligence (AI) can enable a broader, more accurate analysis of quality. More complex machine learning models, such as transformers, now make it possible to analyze all the IEEE 29148 quality attributes, such as completeness and verifiability. It is also possible for AI algorithms to learn complicated relationship patterns among words and sentences. Generative AI further promises to enable automated requirements specification generation.

Modern approaches can use advanced Natural Language Processing (NLP) to measure requirements qualities. For example, techniques such as classification could be used to determine if a given requirement is non-ambiguous, singular, or verifiable. As an extension, transformers and Large Language Models (LLMs) have brought a new way of analyzing text with high context windows and reliable textual analysis, creating new possibilities for requirement quality analysis.

Some current tools such as the Requirements Analysis Tool (RAT) try to measure more complicated quality attributes through non-rule-based techniques. The tool helps utilize modeling techniques to conduct syntactical and semantical analysis of requirements by constructing complicated relationships through the use of user-defined glossaries [6]. Other modern approaches use NLP and machine-learning based approaches, including part of speech (POS) tagging, lexical analysis, semantic analysis, and ML classification [1]. Additional methods have combined rule-based implementations with machine learning. Specifically, a study utilized aggregated text statistics, POS tagging, and readability metrics as features to train a C4.5 decision tree model for requirement quality measurement [7].

Because requirements quality measurement is subjective, other tools incorporate expert input into the training process to allow for a more dynamic and accurate model. The Requirements Quality Analysis (RQA) tool incorporates both quality metrics and input from experts into its training, ultimately achieving better quality benchmarks [8]. Other tools have extended beyond the limitations of traditional NLP-based methods through the utilization of transfer learning. For instance, NoRBERT utilizes transfer learning of the BERT model to classify requirements, showcasing impressive results [9].

Ultimately, transformers seemingly allow for more complicated tasks via transfer learning, while also providing a high context window when generating responses. Due to the highly dynamic and context-specific nature of requirements engineering, transformers are a good fit for most quality-based tasks within requirements documents. Because an LLM is already trained on an enormous set of data, it could potentially analyze the quality of requirements in a way that extends beyond textual heuristics or even other NLP methods. Even though transformers showcase impressive potential, there is a lack of practical tools that utilize modern LLM technology to analyze requirements.

Automated Requirements Quality Measurement (ARQM) tool

Because transformers have provided promising results for difficult NLP-related tasks, there is a motivation to develop a practical tool that utilizes transformers for analyzing requirement quality. We are developing an "Automated Requirements Quality Measurement tool (ARQM)" that serves as the foundation for implementing the newest transformer technology for analyzing requirements. The purpose of the software is to analyze requirements and requirement groups based on the IEEE 29148 standard, while also providing user interaction features that help create a highly dynamic experience. The system as proposed and partially implemented utilizes ChatGPT version 3.5 to analyze individual requirements and groups of requirements at a level that considers the necessary context and practicality of the requirements document. Additionally, the tool will utilize other Sequence to Sequence (Seq2Seq) models, such as T5 and BART, to fine-tune results for the ARQM tool. Ideally, the use of LLMs will allow for a broader and more accurate analysis of overall requirement quality.

The ARQM tool fully integrates with Microsoft Word 2016 or later, allowing the user to declare, validate, and delete requirements through a structured add-in interface. Because of this, the tool will interact with the native Microsoft Word features to provide a better, more holistic experience during the requirements engineering process. Ultimately, the tool will allow a user to 'declare' requirements and requirement groups within their document; from there, the system will analyze the requirements and requirement groups based on the IEEE 29148 requirement quality criteria. The system will also provide rephrasing suggestions to improve the associated quality. In conjunction with that functionality, the system will use transformer-based AI technology to provide a context-friendly analysis of all requirements and requirement groups declared within the document.

In addition to providing basic analytical capabilities for requirements documents, it will also provide extensive features that allow for user feedback during the quality feedback process. For example, ARQM will provide an ability for the user to provide project-specific information and instructions before generating results. Via custom prompts and instructions, the system will allow a user to define critical information, such as budgetary restrictions, before making quality

determinations. It extends that functionality by providing an ability for the user to respond to generated results. For example, the user can provide additional information to the system using messages to generate new quality results. Both features allow the system to maintain and consider user message history and feedback before generating results.

Finally, ARQM utilizes the latest transformer technology to allow for a broader analysis of requirements and requirement groups. The system can potentially analyze requirements on a lexical, syntactical, and semantical basis, providing a strong capability for analyzing requirements quality. Furthermore, it will leverage the power of LLMs to provide a holistic analysis of requirements documents. With all its proposed features, ARQM is a practical, user-friendly and dynamic tool, which allows for a broad and context-specific analysis of requirements documents.

A Bright Future for AI Enabled Requirements Quality Analysis

Due to the increasing complexity of requirements and their associated documentation, existing tools have emerged to help analyze and improve requirements documents. Historical tools have utilized general rules-of-thumb to analyze the quality of requirements, including basic lexical and syntactical analysis. As these tools evolved, newer systems utilized artificial intelligence to provide better analytical capabilities for requirements documents. Advanced AI approaches can capture complex linguistic relationships– allowing for further analysis of requirement quality. Recent advancements with transformers and LLMs have allowed for a broader, context-specific analysis of requirements documents. Unfortunately, the most up-to-date practical tools have not implemented this technology.

The ARQM tool uses modern transformer technology dynamic user feedback to the system to provide a holistic, context-specific analysis of both requirements and requirement groups based on the IEEE 29148 standard. The system extends existing tool functionality through user-specific features and contextual quality analysis, paving the way for a better requirement engineering experience. The tool interfaces with Microsoft Word to provide an easy-to-use interface for declaring, deleting, and analyzing requirements or requirement groups.

Because the system is currently in development, there is still notable work to realize its completion. Primarily, most of the main system features are completed, but there is still work on fine-tuning and prompt engineering to achieve ideal system results when analyzing requirement quality. Further, the integration of the full contextual information provided by users remains a difficult challenge when generating results. Regardless, the ARQM system remains in development and the primary functionality is complete. Further work will outline the results of the ARQM system upon completion.

References

[1] L. Zhao *et al.*, "Natural Language Processing for Requirements Engineering: A Systematic Mapping Study," *ACM Comput. Surv.*, vol. 54, no. 3, Art. no. 3, Apr. 2021, doi: 10.1145/3444689.

[2] IEEE Std 830–1998 (1998) IEEE Recommended Practice for Software Requirements Specifications, Los Alamitos, CA, IEEE Computer Society Press.

[3] "ISO/IEC/IEEE International Standard - Systems and software engineering – Life cycle processes – Requirements engineering," *ISOIECIEEE 291482018E*, pp. 1–104, Nov. 2018, +doi: 10.1109/IEEESTD.2018.8559686.

[4] N. Carlson and P. Laplante, "The NASA automated requirements measurement tool: a reconstruction," *Innov. Syst. Softw. Eng.*, vol. 10, no. 2, Art. no. 2, Jun. 2014, doi: 10.1007/s11334-013-0225-8.

[5] Gardner G., A Review of Automated Tools to Improve Natural Language Requirements, The ITEA Journal of Test and Evaluation 2017; 38: 153-158.

[6] K. Verma and A. Kass, "Requirements Analysis Tool: A Tool for Automatically Analyzing Software Requirements Documents," in *The Semantic Web - ISWC 2008*, A. Sheth, S. Staab, M. Dean, M. Paolucci, D. Maynard, T. Finin, and K. Thirunarayan, Eds., in Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2008, pp. 751–763. doi: 10.1007/978-3-540-88564-1_48.

[7] E. Buchmann and S. Hauser, "Text Mining for Standardized Quality Criteria of Natural-Language IT-Requirements," in *2021 IEEE 29th International Requirements Engineering Conference (RE)*, Sep. 2021, pp. 390–395. doi: 10.1109/RE51729.2021.00043.

[8] V. Moreno, G. Génova, E. Parra, and A. Fraga, "Application of machine learning techniques to the flexible assessment and improvement of requirements quality," *Softw. Qual. J.*, vol. 28, no. 4, Art. no. 4, Dec. 2020, doi: 10.1007/s11219-020-09511-4.

[9] T. Hey, J. Keim, A. Koziolek, and W. F. Tichy, "NoRBERT: Transfer Learning for Requirements Classification," in *2020 IEEE 28th International Requirements Engineering Conference (RE)*, Zurich, Switzerland: IEEE, Aug. 2020, pp. 169–179. doi: 10.1109/RE48521.2020.00028.