Approaches to Evaluating the Quality of IT Project Documentation: A Systematic Literature Review

DOI
https://doi.org/
10.18690/um.fov.2.2025.76

ISBN
978-961-286-963-2

YAUHEN UNUCHAK, TATYANA UNUCHAK

University of Maribor, Faculty of Organizational Sciences, Kranj, Slovenia
yauhen.unuchak@student.um.si, tatyana.unuchak@student.um.si

This research presents the results of a literature review of methods and approaches to assessing the quality of documentation in IT projects. We conducted a systematic review of about 100 publications devoted to the description of tasks and key elements of project documentation. In this paper we tried to systematize and summarize a) the main approaches and methods used to assess the quality of documentation descriptions in IT projects, b) key criteria and metrics that determine the completeness and clarity of documentation descriptions, c) problems encountered in assessing the quality of documentation and textual descriptions of tasks in IT projects. The obtained results provide a basis for our further work aimed at improving the quality and efficiency of project documentation.

Keywords:
IT task description,
software requirements,
text quality metrics,
documentation quality assessment,
IT projects,
natural language processing,
project documentation

University of Maribor Press

1 Introduction

In an era of rapidly evolving technologies, one of the key factors for successful IT project implementation is the quality of project documentation. Documentation serves as the basis for coordination among team members, helps avoid miscommunication, and facilitates effective task management. Deficiencies in documentation quality, such as unclear or incomplete task descriptions, often lead to delays, increased costs, and reduced quality of the final product. Quality documentation helps developers, analysts, and testers to understand requirements more accurately, reduces the risks of errors, and promotes a clearer distribution of responsibilities. The documentation quality assessment is becoming especially relevant in the context of globalization and remote work, where proper understanding of tasks is critical.

Our goal is to investigate the literature sources that touch upon the problem of assessing the quality of IT project documentation, to identify gaps in research and to determine directions for further research. Therefore, the main research questions of our work will be the issues related to assessing the quality of documentation descriptions in IT projects. RQ1 - What approaches and methods are used to assess the quality of documentation descriptions in IT projects? This question aims to summarize the existing approaches and tools that are used to analyze and assess the quality of documentation in IT projects. RQ2 - What are the key criteria that determine the completeness and clarity of documentation descriptions? Here we focus on the characteristics that make task descriptions qualitative and usable. RQ3 - What are the challenges in assessing the quality of documentation and textual task descriptions in IT projects? This question seeks to identify key barriers to effective documentation quality assessment.

In this paper, we present the results of a literature review of methods, approaches and techniques for assessing the quality of task descriptions and key elements of IT project documentation. Approximately 100 articles covering current approaches to assessing the quality of documentation in IT projects were reviewed.

2 Methodology

We used a systematic literature review method (Kitchenham & Charters, 2007) to identify studies investigating the problems of quality assessment of design documentation. We followed a pre-designed research strategy. The research strategy was as follows:

1. Setting the purpose of the study (we defined the purpose in the introduction). We also identified the main directions of the research. We formulated three research questions RQ1, RQ2 and RQ3 (the questions were defined by us in the introduction). When formulating the research questions, we were oriented not only to make a comprehensive analysis on our topic, but also to identify the main gaps in the field of quality assessment of task descriptions and key elements of IT project documentation.
2. Resource selection for literature search. Bibliographic databases such as Web of Science, Scopus and ProQuest Dissertation & Theses were used for the review. These databases were chosen because of their wide coverage of scientific publications and the ability to filter materials by keywords, time of publication and other parameters.
3. Definition of search criteria. Publications were searched using keywords related to the topic of our research study. Some keywords were as follows: "IT task description", 'quality assessment', 'documentation standards', 'software project management', etc. The search included articles published in the period from 1990 to the present time.
4. The implementation of the search itself. We have implemented the search as follows.

4.1. Manual literature search on Web of Science, Scopus and ProQuest Dissertation & Theses. At once we selected articles on these resources by keywords. We ended up with 36 files with the search results for each source. Each file contained the following information about the literature sources found: Publication Type, Authors, Book Authors, Book Editors, Book Group Authors, Author Full Names, Article Title, Source Title, etc. (75 characteristics in total).

4.2. Automatic processing of manual search results. Using a program written in Python we processed the manually found articles.

5. A Data analysis and systematization. All found articles were analyzed for the issues we studied: approaches and methods used to assess the quality of documentation descriptions in IT projects (RQ1), key criteria and metrics determining the completeness and clarity of documentation descriptions (RQ2), problems arising in assessing the quality of documentation and textual descriptions of tasks in IT projects (RQ3). The articles were summarized and systematized by us.

As mentioned above, to implement point 4.2 of our strategy, we wrote a Python program that helped us to choose the most suitable data from the data set obtained in point 4.1. The algorithm of the program was as follows, let's describe it step by step:

- step 1: Integration of all Web of Science, Scopus, ProQuest Dissertation & Theses search results into three Excel files (a different file was created for each search source). As output we got the files webofscience_integration.xls, scopus_integration.xls, ProQuest_integration.xls,
- step 2: Removing columns from the files obtained in the previous step that do not carry important information for our search. As a result, in each of the three files we left only the columns: Title, Abstract, DOI, Cited by, Year,
- step 3: Combining the three files webofscience_integration.xls, scopus_integration.xls, ProQuest_integration.xls into one file, while finding and removing duplicates by DOI. As an output we got the file all_integration.xls,
- step 4: Filtering the file all_integration.xls by our specified keywords. As a result, we got the file all_integration_filtered.xls.

The file all_integration_filtered.xls contained records with the topics set by the filter. There were already much fewer such records than there were originally. Next, we looked through the articles in this file and selected the ones that were suitable for us. The keywords to search for were: “Software requirements”, “Project documentation”, “Structural assessment of requirements”, “Software documentation”, “Text quality evaluation”, “Software requirements” AND “Quality metrics”, “Natural language processing”, “Structural assessment” AND “Requirements”, “Crowd” AND “Clarity description”, “Crowd” AND “Natural language processing", "Text quality" AND "Evaluation" AND "NLP". These keywords were included in searches within the Title and within the Abstract of the article. We tried to use the same filters for Web of Science and for Scopus. English language search filters were also set. The selected literature spans a variety of approaches, from automated analysis tools (e.g., NLP models) to manual evaluation of task descriptions.

3 Results

As described in the Methodology section, our first step was to generate search queries on three search resources Web of Science, Scopus, and ProQuest Dissertation & Theses. The obtained search results are shown in Fig. 1.

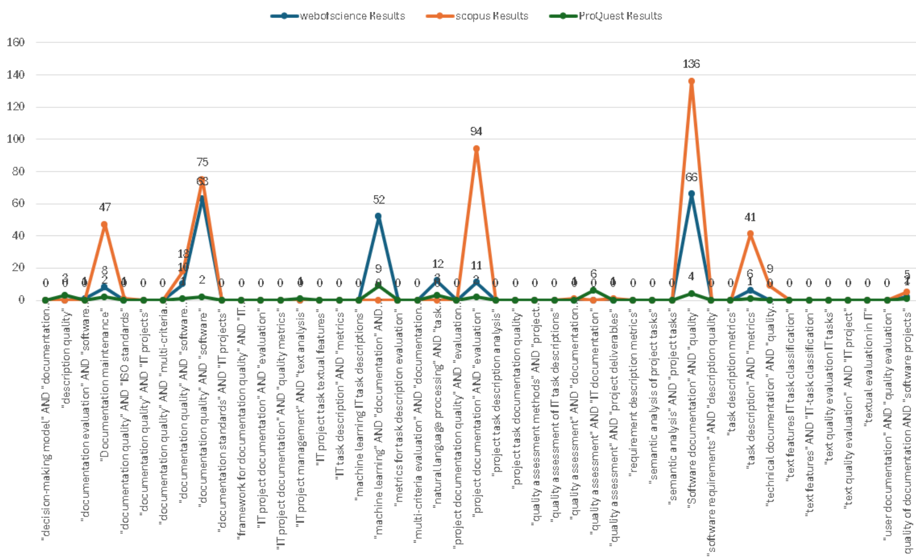

Figure 1: Search results (Y-axis) by queries (X-axis) for three search engines: Web of Science, Scopus, ProQuest Dissertation & Theses

Source: Own

The figure shows that Scopus shows a significantly higher number of results for most search queries. For example, for the queries "documentation quality" AND "software", "project documentation" AND "evaluation" and "software documentation" AND "quality", 75, 94 and 136 publications were found, respectively. Web of Science showed fewer relevant publications, although for some queries (e.g., "software documentation" AND "quality") it provided 66 publications. ProQuest Dissertation & Theses has a limited number of results for all queries. In most cases, the results are either missing or minimal, indicating less complete coverage of the research topic by this platform.

3.1 Approaches and methods used to assess the quality of documentation descriptions in IT projects (RQ1)

The literature review identified a variety of approaches to assessing the quality of documentation and textual descriptions of tasks in IT projects. We divided these approaches into four areas: automated methods, automated criteria and metrics, manual or semi-automated approaches, and standards-based approaches:

- use of automated methods. This can include the use of machine learning for classification tasks (Izadi idr., 2022; Rani idr., 2021), analyzing the vagueness or ambiguity of design problem descriptions (Nouri idr., 2021), estimating the structure of system requirements (Vierlboeck idr., 2024), and using graph metrics to estimate the complexity of system requirements (Aversano idr., 2017b),
- manual and semi-automated methods. These include informal and formal reviews (team discussions of documentation) (Bacchelli idr., 2012; Bettenburg & Premraj, 2024; Plösch idr., 2014), surveys of experts to identify key quality aspects (Plösch idr., 2014). Experts can also be involved to assess text quality criteria, e.g., Consistency (Aversano idr., 2017b; Plösch idr., 2014; Rani idr., 2023), Consistency to Standard (Aversano idr., 2017b; Ulan idr., 2018), Coherence (Pereira idr., 2024; Plösch idr., 2014; Rani idr., 2023; Treude idr., 2020; Zhi idr., 2015),
- use of criteria and metrics for assessing the quality of documentation. They are described in more detail in Table 1,
- using standards-based approaches. This can include standards-based qualitative models (e.g. ISO/IEC 25010 (Aversano idr., 2017b; Ulan idr., 2018) aimed at assessing the requirements captured in standards (functional suitability, reliability, usability and maintainability).

The identified approaches and metrics confirm that document quality assessment is a complex process requiring a combination of automation and expert analysis.

3.2 Key criteria and metrics that determine the completeness and clarity of documentation descriptions (RQ2)

As a result of the study, we found that authors often use similar metrics and quality criteria in their works when assessing the quality of documentation in IT projects, despite the differences in approaches and analysis tools. We systematized these metrics and presented them in the form of a table 1, where each criterion is accompanied by references to literature sources.

It should be noted that most of the criteria discussed, such as readability, clarity, and structure, are used not only in the context of program documentation, but also to evaluate texts in general. For example, they are used in Crowdsourcing (Nouri idr., 2021, 2023; Yang idr., 2024). This suggests that the assessment of documentation quality is based on universal principles related to text properties. Automated readability Index and Fog Index are popular tools for readability assessment because they use objective text parameters (such as, average sentence length, average number of syllables in words) (Lehner, 1993). However, metrics such as Cohesion and Coherence are more often manually assessed (Treude idr., 2020).

Despite their universality, the criteria and metrics highlighted are poorly adapted directly to the project documentation. For example, in the context of software it is also essential to consider such aspects as the detailed description of functional requirements, the presence of use cases and test scenarios, as well as the completeness of the description of API interfaces. Existing metrics do not effectively measure how completely and accurately such elements are presented in the documentation. The criterion of compliance with technical standards requires a comprehensive analysis, including checking whether the structure and content of the documentation comply with the established standards. Automating such checks is difficult due to the diversity of standards and the difficulty in interpreting their requirements. Manual evaluation of some metrics makes the evaluation process labour-intensive, subjective and error-prone.

Table 1: Most frequently mentioned criteria and metrics for the quality of text descriptions

Quality criteria /metric	References to sources
Criteria	
Readability	(Aversano idr., 2017b, 2017a; Lehner, 1993; Nouri idr., 2021; Pereira idr., 2024; Plösch idr., 2014; Rani idr., 2023; Treude idr., 2020; Yang idr., 2024; Zhi idr., 2015)
Clarity	(Bacchelli idr., 2012; Ding idr., 2014; Nguyen idr., 2024; Nouri idr., 2023; Plösch idr., 2014; Treude idr., 2020)
Structuredness	(Aversano idr., 2017b, 2017a; Bettenburg & Premraj, 2024; Plösch idr., 2014)
Completeness	(Aversano idr., 2017b, 2017a; Nguyen idr., 2024; Rani idr., 2023; Zhi idr., 2015)
Specificity	(Izadi idr., 2022; Lehner, 1993; Nouri idr., 2021)
Alignment	(Aversano idr., 2017b; Pereira idr., 2024; Zhi idr., 2015)
Consistency to Standard	(Aversano idr., 2017b; Ulan idr., 2018)
Graphical Support	(Aversano idr., 2017a, 2017b; Vierlboeck idr., 2024; Yang idr., 2024)
Cohesion	(Treude idr., 2020)
Coherence	(Pereira idr., 2024; Plösch idr., 2014; Rani idr., 2023; Treude idr., 2020; Zhi idr., 2015)
Consistency	(Aversano idr., 2017b; Plösch idr., 2014; Rani idr., 2023)
Metrics (used for automatically generated text)	
ROUGE	(Nguyen idr., 2024; Pereira idr., 2024)
BERTScore	(Nguyen idr., 2024)
BLEU	(Nguyen idr., 2024; Pereira idr., 2024)
SummaC	(Nguyen idr., 2024)

Source: own

In view of the above, we believe that we need to focus our efforts on developing specialised metrics and algorithms that will take into account the unique properties of project documentation. These metrics should not only measure the quality of textual elements but also assess the alignment of documentation with functional, technical, and structural requirements. Additionally, creating automated tools to simplify the evaluation process will help reduce subjectivity and improve the efficiency of quality assessment.

3.3 Problems encountered in assessing the quality of documentation and textual descriptions of tasks in IT projects (RQ3)

Based on the literature review conducted within the framework of research question RQ3, it was possible to identify the main problems that the authors highlight in the context of quality assessment of documentation and textual descriptions of tasks in

IT projects. These problems cover both technical and organizational aspects that make it difficult to use and evaluate documentation effectively. We identified the following challenges:

- cluttered with nonnatural language artifacts. Code fragments, stack traces, log results and configuration files increase the size of documentation descriptions (Bettenburg & Premraj, 2024; Hirsch & Hofer, 2022). Sometimes it is necessary that all artifacts are removed from a text because they interfere with its evaluation (Calefato idr., 2019). However, there are works in which authors say that artifacts affect the performance of applying machine learning techniques (Hirsch & Hofer, 2022),
- problems in formatting. Tools such as Markdown allow documentation authors to format their textual descriptions. However, improper use of such tools makes texts difficult to read (Hirsch & Hofer, 2022),
- dependence on manual preprocessing of data. Many authors perform manual preprocessing for natural language, which requires additional zatart resources and is not always economically feasible (Bacchelli idr., 2012; Bettenburg & Premraj, 2024; Izadi idr., 2022),
- lack of clarity in descriptions of textual tasks. These include: the desired solution is not stated, the wording is not easy to understand, potentially important terms are not defined, the presentation format is not specified in sufficient detail, acceptance criteria are not defined (Aversano idr., 2017b, 2017a; Lehner, 1993; Nguyen idr., 2024; Nouri idr., 2023; Rani idr., 2023; Wingkvist idr., 2010; Zhi idr., 2015),
- documentation problems. Unreliable, incomplete, or nonexistent documentation, undocumented changes to the software system, and lack of integrity and consistency in the documentation itself affect the results and quality of its processing (Cummaudo idr., 2024; Rani idr., 2023; Wingkvist idr., 2010; Zhi idr., 2015),
- problems in interpreting automatic text quality assessment metrics such as BLEU and ROUGE (Pereira idr., 2024). Automatic metrics are used for machine translation and summarization tasks, so they do not take into account specific features in IT task descriptions. Their effectiveness for analyzing the quality of project documentation also remains questionable,

- low level of task tagging practices in IT projects. Research shows that only 3% of repositories on GitHub use task tagging, and even in those repositories only 58% of tasks are tagged (Cabot idr., 2024). All of this affects the results of IT task processing.

The identified peculiarities emphasize that current methods of assessing the quality of documentation in IT projects require serious revision and adaptation. The solution to these issues involves the development of specialized tools and metrics adapted to the peculiarities of IT documentation.

4 Conclusion

This study reviewed literature sources addressing the problem of assessing the quality of IT project documentation. We described the methodology and strategy of the study. The review showed a variety of approaches, methods, criteria and metrics used to assess the readability, structure, clarity, etc. of textual descriptions. The problems highlighted in the paper, such as cluttering of texts with artifacts, difficulty in interpreting metrics, lack of clarity of descriptions, etc., are indicative of the challenges faced by current analysis methods.

Our future research should focus on creating solutions that improve the effectiveness of quality assessment tools for project documentation. For example, it is possible to develop a model that can also consider both the completeness of the text description, its clarity, readability, and its relevance to the task at hand in the IT project. The application of machine learning methods for automatic evaluation of project documentation seems to be a promising direction for the realisation of this idea.

References

Aversano, L., Guardabascio, D., & Tortorella, M. (2017a). Analysis of the Documentation of ERP Software Projects. Procedia Computer Science, 121, 423–430. https://doi.org/10.1016/j.procs.2017.11.057

Aversano, L., Guardabascio, D., & Tortorella, M. (2017b). Evaluating the Quality of the Documentation of Open Source Software: Proceedings of the 12th International Conference on Evaluation of Novel Approaches to Software Engineering, 308–313. https://doi.org/10.5220/0006369403080313

Bacchelli, A., Dal Sasso, T., D'Ambros, M., & Lanza, M. (2012). Content classification of development emails. 2012 34th International Conference on Software Engineering (ICSE), 375–385. https://doi.org/10.1109/ICSE.2012.6227177

Bettenburg, N., & Premraj, R. (2024). (PDF) Extracting structural information from bug reports. ResearchGate. https://doi.org/10.1145/1370750.1370757

Cabot, J., Cosentino, V., & Rolandi, B. (2024, oktober 31). Exploring the Use of Labels to Categorize Issues in Open-Source Software Projects. ResearchGate. https://doi.org/10.1109/SANER.2015.7081875

Calefato, F., Lanubile, F., & Vasilescu, B. (2019). A large-scale, in-depth analysis of developers' personalities in the Apache ecosystem. Information and Software Technology, 114, 1–20. https://doi.org/10.1016/j.infsof.2019.05.012

Cummaudo, A., Vasa, R., Grundy, J., & Abdelrazek, M. (2024). Requirements of API Documentation: A Case Study into Computer Vision Services | Request PDF. ResearchGate. https://doi.org/10.1109/TSE.2020.3047088

Ding, W., Liang, P., Tang, A., & van Vliet, H. (2014). Knowledge-based approaches in software documentation: A systematic literature review. Information and Software Technology, 56(6), 545–567. https://doi.org/10.1016/j.infsof.2014.01.008

Hirsch, T., & Hofer, B. (2022). Detecting non-natural language artifacts for de-noising bug reports. Automated Software Engineering, 29(2), 52. https://doi.org/10.1007/s10515-022-00350-0

Izadi, M., Akbari, K., & Heydarnoori, A. (2022). Predicting the objective and priority of issue reports in software repositories. Empirical Software Engineering, 27(2), 50. https://doi.org/10.1007/s10664-021-10085-3

Kitchenham, B., & Charters, S. M. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering (EBSE-2007-01). Department of Computer Science, University of Durham, UK. https://www.researchgate.net/publication/302924724_Guidelines_for_performing_Systematic_Literature_Reviews_in_Software_Engineering

Lehner, F. (1993). Quality control in software documentation: Measurement of text comprehensibility. Information & Management, 25(3), 133–146. https://doi.org/10.1016/0378-7206(93)90036-S

Nguyen, H., Chen, H., Pobbathi, L., & Ding, J. (2024). A Comparative Study of Quality Evaluation Methods for Text Summarization (arXiv:2407.00747). arXiv. https://doi.org/10.48550/arXiv.2407.00747

Nouri, Z., Gadiraju, U., Engels, G., & Wachsmuth, H. (2021). What Is Unclear? Computational Assessment of Task Clarity in Crowdsourcing. Proceedings of the 32nd ACM Conference on Hypertext and Social Media, 165–175. https://doi.org/10.1145/3465336.3475109

Nouri, Z., Prakash, N., Gadiraju, U., & Wachsmuth, H. (2023). Supporting Requesters in Writing Clear Crowdsourcing Task Descriptions Through Computational Flaw Assessment. Proceedings of the 28th International Conference on Intelligent User Interfaces, 737–749. https://doi.org/10.1145/3581641.3584039

Pereira, J., Assumpcao, A., & Lotufo, R. (2024). Check-Eval: A Checklist-based Approach for Evaluating Text Quality (arXiv:2407.14467). arXiv. https://doi.org/10.48550/arXiv.2407.14467

Plösch, R., Dautovic, A., & Saft, M. (2014). The Value of Software Documentation Quality. 2014 14th International Conference on Quality Software, 333–342. https://doi.org/10.1109/QSIC.2014.22

Rani, P., Blasi, A., Stulova, N., Panichella, S., Gorla, A., & Nierstrasz, O. (2023). A decade of code comment quality assessment: A systematic literature review. Journal of Systems and Software, 195, 111515. https://doi.org/10.1016/j.jss.2022.111515

Rani, P., Panichella, S., Leuenberger, M., Di Sorbo, A., & Nierstrasz, O. (2021). How to identify class comment types? A multi-language approach for class comment classification. Journal of Systems and Software, 181, 111047. https://doi.org/10.1016/j.jss.2021.111047

Treude, C., Middleton, J., & Atapattu, T. (2020). Beyond accuracy: Assessing software documentation quality. Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 1509–1512. https://doi.org/10.1145/3368089.3417045

Ulan, M., Hönel, S., Martins, R. M., Ericsson, M., Löwe, W., Wingkvist, A., & Kerren, A. (2018). Quality Models Inside Out: Interactive Visualization of Software Metrics by Means of Joint Probabilities. 2018 IEEE Working Conference on Software Visualization (VISSOFT), 65–75. https://doi.org/10.1109/VISSOFT.2018.00015

Vierlboeck, M., Nilchiani, R., & Blackburn, M. (2024). Natural Language Processing to Assess Structure and Complexity of System Requirements. https://doi.org/10.36227/techrxiv.20818984.v2

Wingkvist, A., Ericsson, M., Lincke, R., & Lowe, W. (2010). A Metrics-Based Approach to Technical Documentation Quality. ResearchGate. https://doi.org/10.1109/QUATIC.2010.88

Yang, K., Qi, H., & Huang, Q. (2024). The impact of task description linguistic style on task performance: A text mining of crowdsourcing contests | Request PDF. ResearchGate. https://doi.org/10.1108/IMDS-03-2021-0178

Zhi, J., Garousi-Yusifoğlu, V., Sun, B., Garousi, G., Shahnewaz, S., & Ruhe, G. (2015). Cost, benefits and quality of software development documentation: A systematic mapping. Journal of Systems and Software, 99, 175–198. https://doi.org/10.1016/j.jss.2014.09.042

About the authors

Tatyana Unuchak is a Master of Science in IT organisation and an expert in the field of web and mobile applications development. Her research interests are related to machine learning and project management process improvement.

Yauhen Unuchak is a Master of Science in IT organisation and an expert in automated testing, software performance testing and machine learning. His research interests include automated testing, quality management systems, corporate information systems, big data and predictive analytics. He is one of the authors of the book "IT-Startup: 10 Tips for Beginners".