ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 2, July 2024

REQUIREMENTS ENGINEERING QUALITY A LITERATURE REVIEW

Rosa Delima[1], Azhari Azhari[3], Khabib Mustofa[3]

[1]Informatika, Universitas Kristen Duta Wacana
[2,3]Ilmu Komputer dan Elektronik, Universitas Gadjah Mada

email: rosadelima@staff.ukdw.ac.id[1], arisn@ugm.ac.id[2], khabib@ugm.ac.id[3]

Abstract

Requirements Engineering Quality (REQ) has a large influence on the success of a software project. A systematic literature review (SLR) is conducted to get complete information about REQ. SLR reviewed 46 relevant publications from 2016 – 2022, sourced from three literature sources: Science Direct, Scopus, and IEEE. Based on the SLR, it is known that, generally, the artifacts processed for REQ are text requirements. The quality standards for REQ that are widely used are ISO/IEEE/IEC 29148 and IEEE 830, while the quality variables that are widely used are correctness, completeness, consistency, and defects/faults found in RE. A number of methods are used to perform automatic REQ. The most widely used method in publications is NLP. This is in line with most artifacts used in REQ, such as text requirements.

Keywords: Requirements Engineering, RE, Validation Model, Requirements Quality, REQ, Requirements Validation.

Received: 24-08-2023 | **Revised:** 24-03-2024 | **Accepted:** 16-05-2024
DOI: https://doi.org/10.23887/janapati.v13i2.53366

INTRODUCTION

The validation stage in Requirements Engineering (RE) is an important stage to ensure the quality of RE. Testing and evaluation are carried out at this stage to determine whether the specifications have met the users' needs, which are defined at the elicitation stage. Quality requirements greatly influence a software project's success [1] [2]. Requirement quality management is an activity that ensures that the results of RE activities meet the requirement standard [2]. In order to be able to produce quality requirements, it is necessary to have good quality for each requirements stage, which begins with elicitation, analysis, and specification.

Based on the literature review conducted, it is known that there are some studies related to Requirements Engineering (RE) [3]. However, no literature study on Requirement Engineering Quality (REQ) has been found. Therefore, this research becomes essential to obtain in-depth information regarding the artifacts, methods, and standards used to measure the quality of requirement engineering outcomes.

Quality assurance on the requirements process is carried out at the validation stage. Validation is based on the artifact specifications formulated. Artifacts will be analyzed, tested using methods/approaches, and adjusted to agreed quality standards. A systematic literature review (SLR) is conducted to get complete information about REQ SLRs are carried out systematically, starting with the Research Question formulation, determination of keywords, literature search, screening paper, data extraction, and continued with analysis and mapping processes [4].

This paper is organized into four parts, starting with an introduction, followed by the methods used in the research in part 2, results and discussion in part 3, and the paper ends with the conclusion in part 4.

METHOD

Literature Review Process

The flow of literature studies carried out can be seen in Figure 1. SLR begins with defining research questions (RQ), which aim to provide a basis for extracting information and focus on the SLR process. After formulating the RQ, the process is continued with a literature search that begins with determining keywords to direct the search process. The search phase will produce a number of papers, which will be in-depth screening to obtain papers relevant to the objectives of the SLR. After obtaining relevant papers, then we extracted information on the paper, analysis, and mapping of information and knowledge related to requirements quality. The formulation of RQ, the search process, and the paper's screening process can be seen in this section.

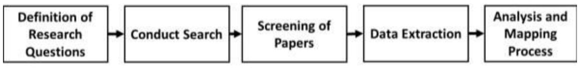
Definition of Research Questions
Conduct Search
Screening of Papers
Data Extraction
Analysis and Mapping Process

Figure 1. Literature review process

Research Questions

The formulation of research questions (RQ) is the initial stage in the SLR. Specific RQ helps give direction and focus to the SLR process. There are four RQs formulated in requirements quality, such as:

- RQ1: What are the artifacts' REQ specifications used in this study?
  Rationale: RQ1 aims to determine the artifact used on the REQ. This is very important to know because the artifact form strongly correlates with the method used in the REQ.
- RQ2: What methods are used for the validation process in the study?
  Rationale: RQ2 aims to obtain information on the type of method used for the validation process on RE.
- RQ3: What are the quality standards and variables used in the study?
  Rationale: RQ3 aims to determine the standards and variables used to measure the quality of RE.

Conduct Search

The search is done through some keywords such as: "requirement quality" or ("requirement engineering" and "quality"). Based on the specified keywords, the literature search process is carried out. The search was carried out for publications from 2016 to 2022 from three literature sources: Science Direct, Scopus, and IEEE. The search uses advanced features, meaning the specified keywords are in the title, abstract, or keywords. The search is focused on articles in the form of journals and proceedings.

Screening Papers

The search using keywords with journal article specifications resulted in 164 papers, and then a screening process was conducted on papers related to inclusion and exclusion factors. Inclusion and exclusion criteria are used to filter papers that are not relevant to RQ [5]. Inclusion is done to filter out the same paper, and exclusion is done to filter irrelevant papers. For the exclusion factor, filtering is done through two stages: 1) examination of abstracts and keywords and 2) abstract reading by the author. The process of abstract examination is carried out by examining the existence of the word "Requirement engineering" and "quality" in the abstract. After passing the first examination, then an abstract reading was carried out by the author to assess the relevance of the paper. Based on the screening results, 46 relevant papers were found. Details of the filtered paper from each stage can be seen in Figure 2.

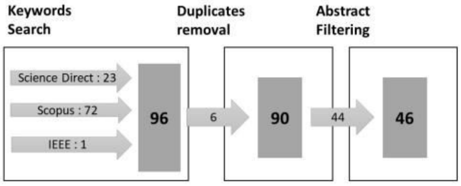

Figure 2. Screening phases and number of selected papers

In Figure 2, it can be seen that the search process based on keywords produced 96 papers, then, it filtered papers by removing duplicate papers, filtering 90 papers, and producing 46 relevant papers.

RESULT AND DISCUSSION

Artifact Specification on REQ

To answer RQ1, a mapping was made regarding the artifact used on the REQ. Based on the SLR, it is known that the artifacts processed for validation are text requirements. These artifacts are natural language (NL) descriptions related to the system's functional requirements. In addition to the text requirement, several publications also use UML in the form of a use case diagram, while user stories become artifacts that are widely used in the agile approach. Other artifacts that are also used include business rules, user interfaces (UI), sequence diagrams, class diagrams, formal specifications, feature descriptions, persona, ontologies, activity diagrams, and state charts. Some publications apply more than one artifact. Even [6] and [7] use four artifacts on REQ. Table 1 is the result of artifact mapping on REQ.

Methods and Tools REQ

RQ3 relates to the methods and tools used in the research that was conducted. In the SLR, information is obtained that each publication uses various methods to validate the RE process. These methods include goal-oriented, test cases, formal methods, quality matrix, Natural Language Processing (NLP), and requirement smell. The most widely used method in publications is NLP. This aligns with most artifacts used in REQ, such as text requirements. Table 4 is a list of REQ methods used in various publications.

Table 1. Artifacts used in REQ

Publication	Artifacts						
	Use Case	**Text/ NL**	**Rule**	**User Stories**	**UI**	**Other**	**Description for other Artifact**
[6]	V	V	V		V		
[2]	V				V		
[8]	V					V	Sequence and class diagram
[9]		V					
[10][11][12][13]		V					
[14]	V						
[15]						V	Formal Specification
[16]		V					
[17]		V					
[18]		V					
[19]			V				
[20]				V			
[21]				V		V	Features
[22]		V					
[23]				V		V	Persona
[24]				V		V	Ontologies
[25]		V					
[26]		V					
[27]				V			
[7]	V	V				V	Activity diagram and State chart
[28]		V					
[29][30][31] [32] [33]		V					
[34]						V	UML Diagram
[35] [36]						V	business process model
[37] [38]	V						
[39]						V	Formal Specification
[40]						V	Scenario
[41]						V	Software req. structural invariants
[42]						V	Semi-formal req. specification doc.

Table 2. REQ standards used in publication

Quality Standard	Publication
ISO/IEEE/IEC-29148	[2] [18] [1] [24] [25] [7] [28] [35]
IEEE 830	[43] [44] [21] [22] [27]
ISTQB standard	[6]
International Requirements Engineering Board (IREB)	[2]
Human Error Taxonomy and Human Error Abstraction Assist (HEAA)	[11]
Safety System Standard : SAE ARP4754A and RTCA DO- 178C	[15]
Software product certification model (SPCM)	[21]
INCOSE Guide for Writing Requirements	[1]
Standard for Aerospace and Defense Industries: ASD-STE100	[22]
Agile Requirements Verification Framework	[45]
Non Functional Requirements (NFRs) Standard: ISO TS 30103 and ISO/IEC 25010	[27]

Table 3. List of quality attributes in REQ

Quality Attribute	Standard	Publications
Correctness	IEEE 830, INCOSE, ASD-STE100, Agile Requirements Verification Framework	[46] [47] [44] [14] [1] [21] [24] [7] [34] [35] [37]
Completeness	ISO/IEEE/IEC-29148, IREB, IEEE 830, ISTQB standard, SPCM, INCOSE, ASD-STE100, Agile Requirements Verification Framework	[6] [46] [47] [44] [14] [16] [1] [21] [22] [23] [24] [25] [7] [38] [42]
Consistency	ISO/IEEE/IEC-29148, IREB, IEEE 830, Agile Requirements Verification Framework	[8] [46] [47] [44] [1] [21] [25] [7] [42] [41]
Traceability	ISO/IEEE/IEC-29148, IREB, ISTQB standard, ASD-STE100	[6] [25]
Defects / Faults	ISTQB standard, HEAA	[6] [9] [11] [17] [26] [45] [12] [13]
Unambiguous	ISO/IEEE/IEC-29148, IREB, IEEE 830, INCOSE, ASD-STE100,	[22] [24] [42]
Non-Vagueness	-	[16] [48]
Comprehensive	ISTQB standard	[6] [47]
Singularity	ISO/IEEE/IEC-29148, IREB, INCOSE, ASD-STE100	[22] [25]
Resource and Dependency Feasibility	ISO/IEEE/IEC-29148, IREB	[47]
Complexity, Coupling, Size	Quality Measure Model	[10]
Safety	SAE ARP4754A and RTCA DO- 178C	[15]
Weak Phrase	-	[16]
Veracity and Variance	ISO/IEEE/IEC-29148	[48]
Uniformity	SPCM, Agile Requirements Verification Framework	[21]
Conformity	SPCM, INCOSE	[21] [22]
Readability	ASD-STE100, ISO/IEC 25010	[22]
clear & concise, necessary, implementation-independent	-	[25]
Verifiable	INCOSE, ASD-STE100	[25]
Feasible	ASD-STE100	[25]
Software Requests and Bug Fixes	-	[49]
Performance		[29]
Conflict		[31] [36]
Verification and validation		[33] [39] [40]

Table 4: Validation method used in REQ

Publication	Method							
	Goal Oriented	Test Cases	Formal	Quality Metric	NLP	Req. Smell	Other	Description of Other Method
[6]	V	V						
[2]		V						
[8][34][33][39][40]			V					
[46][12][13][42]				V				
[9][30][32][12][13]							V	Machine Learning
[47][10][49]							V	Statistical Analysis
[11]							V	Fault Checklist
[14]		V				V	V	Anti-Patterns

[46][12] [13][42]				V				

Table 4: Validation method used in REQ (Continue)

Publication	Method							
	Goal Oriented	**Test Cases**	**Formal**	**Quality Metric**	**NLP**	**Req. Smell**	**Other**	**Description of Other Method**
[15]			V					
[17]						V		
[18]						V		
[20]	V							
[22]					V			
[23]				V			V	INVEST Grid
[25]							V	Expert Judgment
[26]					V			
[45]					V			
[27]							V	Process Pattern
[7]				V	V		V	Rule Based
[28]					V			
[29]	V						V	Intelligence Agent
[30] [31]							V	Genetic Algorithm
[35] [31] [42]							V	Rule-based
[35]							V	Ontology
[12] [13]							V	Clustering Algorithm
[38]	V							
[40]							V	Scenario
[41]							V	Set Theory
[37]							V	Classification algorithm; decision model
[36]							V	BPMN

Discussion

The application of artifacts is related to the system development model used. Some papers explicitly stated the application of the agile model [6] [18] [20] [23] [45] [28] [21], V model [6] [1] [48], formal model [8][15] [18] [25] [7] [24], predictive or traditional model [16], and unified model [23], while some papers did not specify the system development used or intended.

Table 3 describes REQ attributes and standards that have attributes and publications that use attributes. Three attributes are most widely used in research: Correctness, Completeness, and Consistency (3C). Correctness is a quality attribute that can be accessed by tracking a set of needs that stakeholders have agreed upon; then, an examination will be made of whether the needs are correctly stated and reflect what stakeholders have stated and desired. Eight publications use the correctness for the validation process [46] [47] [44] [14] [1] [21] [24] [7]. Completeness is examined based on the conformity between agreed needs and the needs expressed by each stakeholder after negotiations. The attributes used by thirteen publications are such as [6] [46] [47] [44] [14] [16] [1] [21] [22] [23] [24] [25] [7]. The third attribute, consistency, is identified by checking all needs after negotiation. Every need will be checked to see whether it meets other needs. There should be consistency between the number of all needs and the number of needs so that the value of the consistency attribute is obtained [47]. Eight publications use consistency attributes such as [8] [46] [47] [44] [1] [21] [25] [7] and all three attributes are used together in six publications [46] [47] [44] [1] [21] [7].

In the previous research, tools were developed to support the application of methods including System Quality Analyzer (SQA) [46], ScopeMaster® [43], ConQat [17], Cassbeth [16], Requirement Quality Suite (RQS) [1], Respecify [26], and Automatic Quality User Story Artisan (AQUSA) [45]. SQA is a tool that supports quality analysis requirements through correctness, completeness, and consistency metrics [46]. ScopeMaster® is a tool developed by Albion Technology Ltd. It is used for defect detection and measuring functional requirements. This tool accepts input from textual requirements [43]. ConQat, Cassbeth, RQA, Respecity, and AQUSA are tools that receive input from the textual requirements and user stories for AQUSA. This tool is used to analyze quality requirements. In addition to special tools for analyzing quality requirements, several publications also use general tools, such as analysis tools for graph [20], github [48], and statistics [44].

Based on the SLR conducted, it is known that several publications contribute to the development of models, methods, frameworks, approaches, or tools. Table 5 is a list of contributions given to the research conducted.

CONCLUSION

Requirements engineering quality (REQ) modeling is essential to support the success of a software engineering project to get complete information about the requirements quality for Systematic Literature Review (SLR). The papers on SLR are sourced from Science Direct, Scopus, and IEEE. Based on the search and filtering results, 46 relevant papers were obtained from 2016 to early 20. The results of extracting information from the systematic literature review were categorized into three categories: artifacts, standards, and quality variables, and methods to support REQ.

Based on the SLR, it is known that the artifacts processed for REQ are text requirements. These artifacts are natural language descriptions related to the system's functional requirements. In addition to the text requirement, several publications also use UML in the form of a use case diagram. At the same time, user stories become artifacts that are widely used in the agile approach.

Table 5: List of Contributions in REQ

Publi-cation	Type	Name
[2]	Model	Activity-Based RE artifact Quality Models (ABRE-QMs)
[8]	Model	A Multiview Formal Model of Use Case Diagrams Using Z Notation
[46]	Method	Method for analysis of requirement quality evolution
[9]	Method	Supervised learning classifiers for automated verifying the reviews obtained during requirements inspections
[11]	Method	Error-Abstraction and Inspection (EAI) Method
[15]	Tool	Requirements Quality Suite Tool
[17]	Tool	Qualicen Requirements Scout tool
[18]	Tool	Requirements Smells (Smella) Tool
[21]	Framework	Framework for just-in-time Requirements
[24]	Framework	Framework for Domain Ontologies for Requirement Specification and Quality
[45]	Framework and Tool	The Quality User Story framework and AQUSA (automatic Quality User Story) Tool
[29]	Framework	Framework for evaluating Complex adaptive systems
[30]	Method	Methodology for requirements quality classification
[35]	Framework	Rule-based ontology framework (ROF)
[31]	Method	Method to detect conflicts and resolve them.
[32]	Approach	An approach based on cloud service for automated detection of quality requirements
[12] [13]	Approach	MOKSA (Mapping of key phrases to SRS Approach)
[33]	Framework	The framework is based on formal methods for developing safety-critical systems, from requirements analysis to code generation.
[38]	Approach	An approach Goal2UCM to transform an iStar 2.0 model into a use case model
[39]	Approach	A systematic engineering approach, named Formal Requirement Engineering Platform in Aircraft (FREPA)
[42]	Model	SRCM –a semi-formal requirements representation model based
[40]	Tools	Tool BeSoS that supports the iterative and behavior-driven specification of requirements in a Systems of Systems -SoS context.
[41]	Approach	A Domain-Specific Language (DSL) based on Set Theory for requirement analysts.
[37]	Approach	An automated test case classification and prioritization approach that supports the use of case-driven testing in product lines.

[36]	Framework	Business Process Modeling Notation -BPMN-based framework

The quality standards for REQ that are widely used are ISO/IEEE/IEC 29148 and IEEE 830, while the quality variables that are widely used are correctness, completeness, consistency, and defects/faults found in RE. A number of methods are used to perform automatic REQ. The most widely used method in publications is NLP. This is in line with most artifacts used in REQ, such as text requirements. Other methods that are quite widely used are test cases, quality matric, requirements smell, goal-oriented, and formal methods.

ACKNOWLEDGMENT

Special thanks to the Computer Science and Electronics Department Universitas Gadjah Mada and the Faculty of Information Technology Universitas Kristen Duta Wacana for providing the facilities and funding to publish this article.

REFERENSI

[1] E. Gallego *et al.*, “Requirements Quality Analysis: A Successful Case Study in the Industry,” in *Proceedings of the Seventh International Conference on Complex Systems Design & Management, CSD&M Paris 2016*, 2016, pp. 187–201. doi: 10.1007/978-3-319-49103-5.

[2] H. Femmer and A. Vogelsang, “Requirements Quality is Quality in Use,” in *IEEE Software*, 2018, pp. 1–7. doi: 10.1109/MS.2018.110161823.

[3] R. Delima, Khabib Mustofa, and Anny Kartika Sari, “Automatic Requirements Engineering: Activities, Methods, Tools, and Domains – A Systematic Literature Review,” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 7, no. 3, pp. 564–578, 2023, doi: 10.29207/resti.v7i3.4924.

[4] B. Kitchenham, “Guidelines for performing Systematic Literature Reviews in Software Engineering,” 2007. doi: 10.1145/1134285.1134500.

[5] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, “Systematic Mapping Studies in Software Engineering,” in *Proceedings of the 12th international conference on Evaluation and Assessment in Software Engineering*, 2007, vol. 80, no. 2, pp. 68–77. doi: 10.1142/S0218194007003112.

[6] A. Beer and M. Felderer, “Measuring and improving testability of system requirements in an industrial context by applying the goal question metric approach,” in *ACM/IEEE 5th International Workshop on Requirements Engineering and Testing Measuring*, 2018, pp. 25–32. doi: 10.1145/3195538.3195542.

[7] A. Rauh, W. Golubski, and S. Queins, “Semantic Integration of System Specifications to Support Different System Engineering Disciplines,” in *12th IEEE International Symposium on Service-Oriented System Engineering, SOSE 2018 and 9th International Workshop on Joint Cloud Computing, JCC 2018*, 2018, pp. 53–62. doi: 10.1109/SOSE.2018.00016.

[8] K. El Miloudi and A. Ettouhami, “A Multiview Formal Model of Use Case Diagrams Using Z Notation: Towards Improving Functional Requirements Quality,” *J. Eng. (United States)*, vol. 2018, 2018, doi: 10.1155/2018/6854920.

[9] M. Singh, V. Anu, G. S. Walia, and A. Goswami, “Validating Requirements Reviews by Introducing Fault-Type Level Granularity : A Machine Learning Approach,” in *11th Innovations in Software Engineering Conference*, 2018, pp. 1–11. doi: 10.1145/3172871.3172880.

[10] V. Antinyan and M. Staron, “Rendex : A method for automated reviews of textual requirements,” *J. Syst. Softw.*, vol. 131, pp. 63–77, 2017, doi: 10.1016/j.jss.2017.05.079.

[11] V. Anu, G. Walia, W. Hu, J. C. Carver, and G. Bradshaw, “Using a Cognitive Psychology Perspective on Errors to Improve Requirements Quality: An Empirical Investigation,” in *27th International Symposium on Software Reliability Engineering, ISSRE*, 2016, pp. 65–76. doi: 10.1109/ISSRE.2016.41.

[12] M. Singh and G. Walia, “Automating Key Phrase Extraction from Fault Logs to Support Post-Inspection Repair of Software Requirements,” 2021.

[13] M. Singh and G. S. Walia, “Automated mapping of fault logs to SRS requirements using key-phrase extraction,” in *Proceedings of the 2021 ACMSE Conference - ACMSE 2021: The Annual ACM Southeast Conference*, 2021, pp. 138–145. doi: 10.1145/3409334.3452043.

[14] A. Beer, M. Junker, H. Femmer, and M. Felderer, “Initial investigations on the influence of requirement smells on test-case design,” in *IEEE 25th International Requirements Engineering Conference Workshops, REW 2017*, 2017, pp. 323–326. doi: 10.1109/REW.2017.43.

[15] M. Böschen, R. Bogusch, A. Fraga, and C. Rudat, “Bridging the gap between natural language requirements and formal specifications,” in *22nd International Conference on Requirements Engineering (CEUR): Foundation for Software Quality (REFSQ 2016)*, 2016, vol. 1564.

[16] J. L. Dargan, J. S. Wasek, and E. Campos-Nanez, “Systems performance prediction using requirements quality attributes classification,” *Requir. Eng.*, vol. 21, no. 4, pp. 553–572, 2016, doi: 10.1007/s00766-015-0232-4.

[17] H. Femmer, “Requirements quality defect detection with the qualicen requirements scout,” in *24nd International Conference on Requirements Engineering (CEUR): Foundation for Software Quality (REFSQ 2018)*, 2018, vol. 2075.

[18] H. Femmer, D. Méndez Fernández, S. Wagner, and S. Eder, “Rapid quality assurance with Requirements Smells,” *J. Syst. Softw.*, vol. 123, pp. 190–213, 2017, doi: 10.1016/j.jss.2016.02.047.

[19] H. Femmer, M. Unterkalmsteiner, and T. Gorschek, “Which requirements artifact quality defects are automatically detectable? A case study,” in *IEEE 25th International Requirements Engineering Conference Workshops, REW 2017*, 2017, pp. 400–406. doi: 10.1109/REW.2017.18.

[20] J. Gillain, I. Jureta, and S. Faulkner, “Planning Optimal Agile Releases via Requirements Optimization,” in *IEEE 24th International Requirements Engineering Conference Workshops (REW)*, 2016, pp. 10–16. doi: 10.1109/REW.2016.016.

[21] P. Heck and A. Zaidman, “A framework for quality assessment of just-in-time requirements: the case of open source feature requests,” *Requir. Eng.*, vol. 22, no. 4, pp. 453–473, 2017, doi: 10.1007/s00766-016-0247-5.

[22] J. Kang and J. Park, “Data anonymization for requirements quality analysis: A reproducible automatic error detection task,” in *LREC 2018 - 11th International Conference on Language Resources and Evaluation*, 2019, pp. 4432–4436. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85059913207&partnerID=40&md5=395352a706d67cf9d4dce4551224b72b

[23] N. T. Khanh, J. Daengdej, and H. H. Arifin, “Human stories: a new written technique in agile software requirements,” in *Proceedings of the 6th International Conference on Software and Computer Applications - ICSCA ’17*, 2017, pp. 15–22. doi: 10.1145/3056662.3056680.

[24] C. Kücherer, “Use of domain ontologies to improve requirements quality,” in *CEUR Workshop Proceedings -REFSQ workshop*, 2017, vol. 1796.

[25] P. Kummler, L. Vernisse, and H. Fromm, “How good are my requirements? - A new perspective on the quality measurement of textual requirements -,” in *International Conference on the Quality of Information and Communications Technology, QUATIC 2018*, 2018, pp. 156–159. doi: 10.1109/QUATIC.2018.00031.

[26] M. Ledger, “A Demonstration of Respecify: A Requirements Authoring Tool Harnessing CNL,” in *IEEE 25th International Requirements Engineering Conference, RE 2017*, 2017, pp. 472–473. doi: 10.1109/RE.2017.85.

[27] P. V Nistala, K. V Nori, and S. Natarajan, “Process patterns for requirement consistency analysis,” in *the 21st European Conference on Pattern Languages of Programs*, 2016, pp. 1–11. doi: 10.1145/3011784.3011809.

[28] C. Reytérou, “Requirements quality in the incremental design processes: Problems and perspectives.,” in *CEUR Workshop Proceedings - 2nd Workshop on Continuous Requirements Engineering (CRE)*, 2016, vol. 1564.

[29] A. Douglas, T. Mazzuchi, and S. Sarkani, “A stakeholder framework for evaluating the -ilities of autonomous behaviors in complex adaptive systems,” *Syst. Eng.*, vol. 23, no. 5, pp. 633–655, 2020, doi: 10.1002/sys.21555.

[30] D. Adanza Dopazo, V. Moreno Pelayo, and G. Génova Fuster, “An automatic methodology for the quality enhancement of requirements using genetic algorithms,” *Inf. Softw. Technol.*, vol. 140, no. July, 2021, doi: 10.1016/j.infsof.2021.106696.

[31] M. Aldekhail and M. Almasri, “Intelligent identification and resolution of software requirement conflicts: Assessment and evaluation,” *Comput. Syst. Sci. Eng.*, vol. 40, no. 2, pp. 469–489, 2021, doi: 10.32604/CSSE.2022.018269.

[32] R. R. R. Merugu and S. R. Chinnam, “Automated cloud service based quality requirement classification for software requirement specification,” *Evol. Intell.*, vol. 14, no. 2, pp. 389–394, 2021, doi: 10.1007/s12065-019-00241-6.

[33] N. K. Singh, M. Lawford, T. S. E. Maibaum,

and A. Wassyng, “A formal approach to rigorous development of critical systems,” *J. Softw. Evol. Process*, vol. 33, no. 4, 2021, doi: 10.1002/smr.2334.

[34] A. Z. O. Salmon, P. M. G. del Foyo, and J. R. Silva, “A Formal Approach to Requirements Engineering of Automated Systems: Facing the Challenge for New Automated Systems,” *J. Control. Autom. Electr. Syst.*, vol. 32, no. 4, pp. 815–829, 2021, doi: 10.1007/s40313-021-00731-y.

[35] A. P. Yanuarifiani, F.-F. Chua, and G.-Y. Chan, “Performance measurement of a rule-based ontology framework (ROF) for auto-generation of requirements specification,” *Int. J. Inf. Technol. Syst. Approach*, vol. 15, no. 1, 2022, doi: 10.5220/0005329904030410.

[36] Q. Ramadan, D. Strüber, M. Salnitri, J. Jürjens, V. Riediger, and S. Staab, “A semi-automated BPMN-based framework for detecting conflicts between security , data-minimization , and fairness requirements,” *Softw. Syst. Model.*, vol. 19, pp. 1191–1227, 2020, doi: 10.1007/s10270-020-00781-x.

[37] I. Hajri, A. Goknil, F. Pastore, and L. C. Briand, “Automating system test case classification and prioritization for use case-driven testing in product lines,” *Empir. Softw. Eng.*, vol. 25, no. 5, pp. 3711–3769, 2020, doi: 10.1007/s10664-020-09853-4.

[38] Y. Yang, Y. Bok, Z. Yang, E. Sheriff, and T. Li, “Goal2UCM: Automatic Generation of Use Case Model from iStar Model,” in *Proceedings of the 14th International iStar Workshop*, 2021, pp. 21–27.

[39] J. Feng *et al.*, “FREPA: An automated and formal approach to requirement modeling and analysis in aircraft control domain,” in *ESEC/FSE 2020 - Proceedings of the 28th ACM Joint Meeting European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2020, pp. 1376–1386. doi: 10.1145/3368089.3417047.

[40] C. Wiecher and J. Greenyer, “BeSoS: A Tool for Behavior-driven and Scenario-based Requirements Modeling for Systems of Systems,” in *CEUR Workshop Proceedings*, 2021, vol. 2857.

[41] M. Vidal, T. Massoni, and F. Ramalho, “A domain-specific language for verifying software requirement constraints,” *Sci. Comput. Program.*, vol. 197, p. 102509, 2020, doi: 10.1016/j.scico.2020.102509.

[42] M. Osama, A. Zaki-Ismail, M. Abdelrazek, J. Grundy, and A. Ibrahim, “SRCM: A semi formal requirements representation model enabling system visualisation and quality checking,” in *MODELSWARD 2021 - Proceedings of the 9th International Conference on Model-Driven Engineering and Software Development*, 2021, pp. 278–285. doi: 10.5220/0010271202780285.

[43] E. Ungan, C. Hammond, and A. Abran, “Automated COSMIC Measurement and Requirement Quality Improvement Through ScopeMaster ® Tool,” in *CEUR Workshop Proceedings*, 2018, pp. 1–13.

[44] F. Anwar and R. Razali, “Stakeholders selection model for software requirements elicitation,” *Am. J. Appl. Sci.*, vol. 13, no. 6, pp. 726–738, 2016, doi: 10.3844/ajassp.2016.726.738.

[45] G. Lucassen, F. Dalpiaz, J. M. E. M. van der Werf, and S. Brinkkemper, “Improving agile requirements: the Quality User Story framework and tool,” *Requir. Eng.*, vol. 21, no. 3, pp. 383–403, 2016, doi: 10.1007/s00766-016-0250-x.

[46] E. Parra, J. Luis, D. Vara, and L. Alonso, “Poster : Analysis of Requirements Quality Evolution,” in *ACM/IEEE 40th International Conference on Software Engineering: Companion Proceedings Poster:*, 2018, pp. 199–200.

[47] S. Ahmad and S. A. Asmai, “Measuring software requirements quality following negotiation through empirical study,” *Int. J. Appl. Eng. Res.*, vol. 11, no. 6, pp. 4190–4196, 2016.

[48] D. Gopal and K. Lyytinen, “Effects of Social Structures in Requirements Quality of Open Source Software Project Development,” in *Proceedings of the Thirty Eighth International Conference on Information Systems (ICIS)*, 2017, pp. 1–11. [Online]. Available: http://aisel.aisnet.org/icis2017/DigitalPlatforms/Presentations/7

[49] D. Gopal, “Effect of social networks on requirements engineering in open source projects,” in *22nd Americas Conference on Information Systems, AMCIS 2016:*, 2016, pp. 1–5. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84987629331&partnerID=40&md5=1d155452d02f71347845d2cfbe2cbfcb