
Journal of Informatics and Web Engineering

Vol. 1 No. 2 (September 2022)

eISSN: 2821-370X

A Systematic Review on Non-Functional Requirements Documentation in Agile Methodology

Steven Loh Mun Keong¹, Zarina Che Embi^{2*}

^{1,2}Multimedia University, Malaysia

*corresponding author: zarina.embi@mmu.edu.my

Abstract - This systematic literature review studies and summarizes findings of requirements documentation of non-functional requirements practiced by agile software development teams. It identifies current practices and existing gaps when agile software teams discuss and implement non-functional requirements, and the current methods used to document non-functional requirements. Our aim is to identify available evidence on current practices and gaps in documenting non-functional requirements. The review was conducted by searching major databases for publications between 2018 and 2022. The inclusion and exclusion criteria as well as quality assessment scoring criteria were subsequently established. Results show that common themes in the practices and gaps in requirements documentation by agile teams do intersect although they are scarce. This review has the benefit for practitioners to have a better approach to document the requirements that may result in a more quality software product.

Keywords— Agile methodology, Requirements documentation, Non-functional requirements, Practices and gaps, Systematic review

Received: 27 June 2022; Accepted: 18 July 2022; Published: 16 September 2022

I. INTRODUCTION

The agile approach of managing software teams is booming in popularity across the software development industry. It has many advantages over traditional approaches such as waterfall model, such as iterative feedback and constantly in a shippable state of product, faster time to market and adaptive to requirements change. However, agile methodology does not put focus in documentation and in this literature review, the lack of documentation in requirements is studied and analysed, specifically in the area of non-functional requirements (NFR).

Agile methodology put emphasis in communication and collaboration among members within and among agile software development (ASD) teams. Requirements are usually briefly communicated as user story, and all parties rely on collaboration to finalize the details of requirements and expectations of the deliverables. The main characteristic of a user story is that it is presented in a format to express the intent of a user of a system to perform a desired task or



Journal of Informatics and Web Engineering

<https://doi.org/10.33093/jiwe.2022.1.2.2>

© Universiti Telekom Sdn Bhd. This work is licensed under the Creative Commons BY-NC-ND 4.0 International License.

Published by MMU Press. URL: <https://journals.mmupress.com/jiwe>

output. The user story template is rarely used to indicate the non-functional expectations (or quality component) of a software system.

This poses several drawbacks such as miscommunication, traceability, validation, effort estimation, project costing and so on. Moreover, the constant nature of teams' rotation and headcount turnover means that the details and agreement over NFRs are usually lost over time.

This literature review will look at the researches that have been carried out to study this issue in the aspect of impacts and challenges due to lack of documentation in NFRs. It will also look into existing practices and proposed ways to overcome this issue.

Section II of this paper elaborates the search process. Section III presents the findings and summary of the literatures reviewed and Section IV discusses the content of the search results and their limitation, as well as the limitations of this literature review.

II. METHODS

This systematic literature review is conducted using the original guidelines proposed by [1] as reference. The steps undertaken for this research are as presented in the following sections.

A. Research Questions

The objectives of this systematic literature review are formulated into research questions (RQ) indicated as below:

- RQ1. In the context of requirements engineering, what are the gaps that exist when proper requirements documentation practice is replaced with collaboration?
- RQ2. How are the challenges identified in RQ1 impacting ASD teams and their software delivery goals?
- RQ3. What current methods do ASD teams use to overcome the challenges identified in RQ1?
- RQ4. What are the limitations of the current researches?

All review materials used in this research pertaining to RQ1-RQ4 above are literatures published in the last five years (since 2018). The software development space is a very fast evolving industry. This is to ensure that this review is based on up-to-date industry practices and also to take into consideration the newest tools which are made available to facilitate ASD teams in their requirements engineering practices.

B. Search Process

The search process was a manual search of specific conference proceedings and journal papers since 2018. Due to time constraint and limited review capability, this systematic literature review is limited to a preliminary search result from the first few pages from Google Scholar as shown in Table 1. Each journal or conference proceeding was reviewed based on the last five years of publication date, its potential relevance and journals and conferences from well-known publishers like Institute of Electrical and Electronics Engineers (IEEE), Springer and Association of Computing Machinery (ACM) were selected. The following keywords were combined when performing the search to achieve higher search result accuracy: "non-functional requirements", "quality requirements", "documentation", "agile" and "rapid development". Subsequently, the literature title, abstract and conclusion were analysed to determine its relevance. It would be selected if the topic discussion included NFR and agile methodologies, the current status-quo, existing practices by ASD teams and the challenges and consequences faced by ASD teams today.

Table 1. Search queries (SQ) on Google Scholar

No	Search queries
1	“non-functional requirements” documentation agile
2	“quality requirements” documentation agile
3	documentation “non-functional requirements” “rapid development”
4	documentation “quality requirements” “rapid development”

C. Inclusion and exclusion criteria

The search results are filtered further based on certain inclusion and exclusion criteria (see Table 2).

Inclusion criteria:

- I1. Literature is related to requirements engineering.
- I2. The paper discusses NFR from the perspective of documentation and traceability.
- I3. In the context of agile manifesto and methodology.
- I4. The literature describes current practices and challenges with respect to NFR documentation and agile.

Exclusion criteria:

- E1. Publications that do not discuss about non-functional or quality requirements.
- E2. Requirements engineering not in the context of agile manifesto and methodology.
- E3. Published before year 2018.
- E4. Search results that are not published in IEEE, Springer or ACM.
- E5. Duplicate search results.
- E6. Non-English literatures.
- E7. Exclusive content.

Table 2. Inclusion and exclusion criteria filter result

Publisher	Result count	Inclusion filtered	Exclusion filtered
IEEE	49	27	11
Springer	33	16	4
ACM	31	14	5
Others	102	59	0

D. Quality Assessment

The literatures obtained after filtering the search results are reviewed in detail and quality is rated based on a set of quality assessment.

QA1. Is the literature relevant to requirements documentation?

QA2. Does the literature research about requirements documentation in the context of agile methodologies?

QA3. Does the literature discuss the current practices of NFR documentation in ASD teams?

QA4. Does the literature assess the challenges of non-functional requirements documentation in ASD teams and proposes solutions to the challenges?

The quality is then rated with scores Y=1.0, P=0.5 and N=0.0.

E. Data Collection

There are several types of research papers related to the topic that are focused by this systematic literature review. The following data and metadata of the selected papers are obtained:

- Full reference of the literature material.
- Type of research
- Research domain
- Authors
- Summary, research questions and answers of the research
- Quality evaluation
- Whether the research is relevant to requirements engineering and specific to requirements documentation in ASD.

F. Data Analysis

The results are analysed and categorically grouped to answer the research questions:

- The current practices of documenting NFR in agile environments (RQ1).
- Challenges faced by ASD teams due to lack of documentation during requirements engineering phase (RQ1).
- The impact and consequences to the overall software delivery goals in the absence of adequate requirements documentation (RQ2).
- The techniques used by ASD teams to overcome the lack of documentation (RQ3).
- The limitations of the research paper (RQ4).

III. RESULTS

This section presents the results of the literature search activity that are subsequently used to conduct this systematic literature review.

A. Search Results

A total of twenty literatures were identified from the manual search, all of which focused in requirements documentation. The literatures discuss the nature of documenting non-functional requirements in software development. Table 3 depicts the search results with the publication year and the main topic of content that are used in this review. The breakdown of the search results according to article type is shown in Figure 1.

Table 3. Search Results

ID	Author	Year	Main area
[2]	A. Alhazmi and S. Huang	2020	Requirements engineering and documentation
[3]	A. Jarzębowski and P. Weichbroth	2021	Non-functional requirements in agile
[4]	B. Boehm, D. Rosenberg and N. Siegel	2019	Quality factors in agile
[5]	B. Habib and R. Romli	2021	Requirements engineering and documentation
[6]	C. Werner, Z. S. Li, N. Ernst and D. Damian	2020	Non-functional requirements and documentation
[7]	Franch, X. et al.	2018	Elicitation, assessment and documentation
[8]	Heck, P., Zaidman, A	2018	Quality factors in agile
[9]	J. Pasuksmit, P. Thongtanunam and S. Karunasekera	2021	Non-functional requirements in agile
[10]	Jarbele C. S. Coutinho, Wilkerson L. Andrade, and Patrícia D. L. Machado	2019	Requirements engineering and documentation
[11]	Kamran Ali Memon, Xia Xiaoling, and Habiba Halepoto	2019	Quality factors in agile
[12]	M. Shafiq and U. s. Waheed	2018	Elicitation, assessment and documentation
[13]	M. Younas et al.	2020	Non-functional requirements in agile
[14]	Methinee Amorndettawin and Twittie Senivongse	2019	Non-functional requirements in agile
[15]	Nurbojatmiko, Eko K. Budiardjo, and Wahyu C. Wibowo	2018	Requirements engineering and documentation
[16]	Oriol, M., Martínez-Fernández, S., Behutiye, W. et al.	2020	Quality factors in agile
[17]	Robiolo, G., Scott, E., Matalonga, S., Felderer, M.	2019	Non-functional requirements and documentation
[18]	S. Tariq and S. M. Cheema	2021	Requirements engineering and documentation
[19]	W. Alsaqaf, M. Daneva and R. Wieringa	2018	Non-functional requirements in agile
[20]	W. Behutiye, P. Rodríguez, M. Oivo, S. Aaramaa, J. Partanen and A. Abhervé	2020	Non-functional requirements in agile
[21]	Woubshet Behutiye, Pertti Seppänen, Pilar Rodríguez, and Markku Oivo	2020	Requirements engineering and documentation

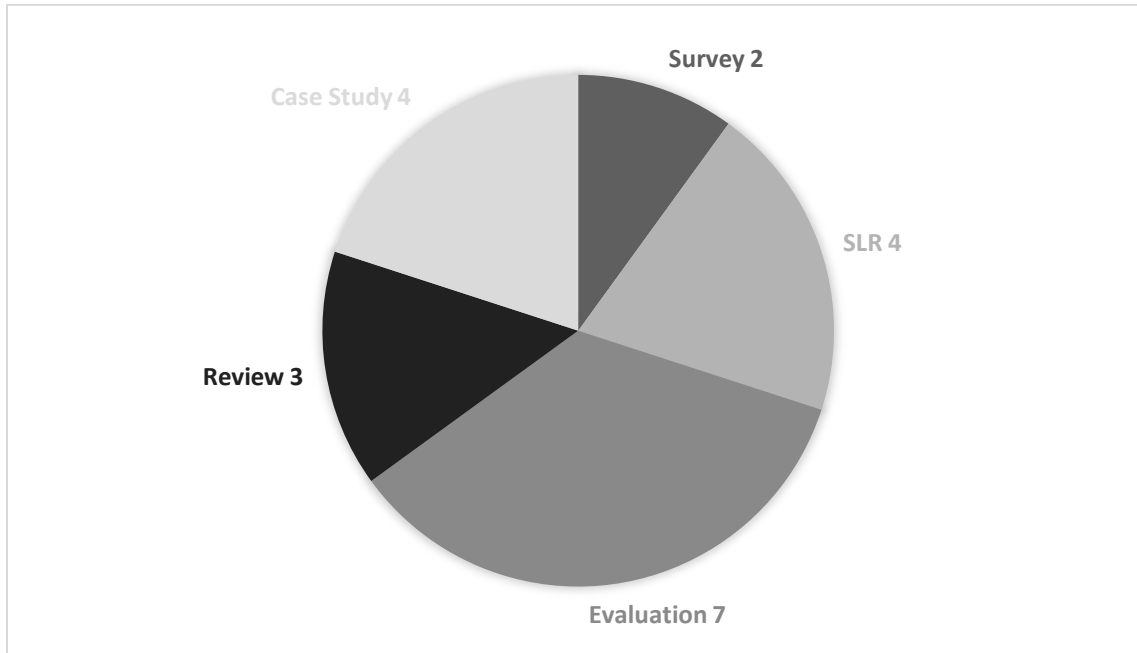


Figure 1: Search results by article type

B. Quality Factors

The contents of literatures are reviewed and the relevance are cross-referenced using the quality assessment defined in section II-D. The quality of literatures obtained from the search results are assessed and evaluated as presented in Table 4 and Figure 2.

The assessment contains the highest score of 4.0 and the lowest score of 1.5. Two papers (10%) scored the highest score of 4.0, five papers (25%) scored 3.5 and 3.0 respectively, three papers (15%) scored 2.5, two papers (10%) scored 2.0 and three papers (15%) scored 1.5. Since all papers scored a minimum of 1.5 and most papers obtained 3.0 and above, it can be assumed that the selected papers have the overall quality criteria for review.

IV. DISCUSSION

This section discusses and reviews existing researches to identify the existing practices of documenting NFRs affecting the majority of software teams practicing agile methodology (RQ1). As a result of lack of investments to document NFRs, the challenges and consequences are identified (RQ2) and the available strategies to mitigate this gap (RQ3) will also be discussed. Finally, the discussion will provide a critique on the limitations of this literature review and prior research work.

Table 4. Quality evaluation of literatures from search results

ID	Article type	QA1	QA2	QA3	QA4	Total Score
[2]	Survey	P	N	N	Y	1.5
[3]	SLR	Y	Y	Y	P	3.5
[4]	Evaluation	P	P	Y	N	2.0

[5]	Review	Y	Y	Y	Y	4.0
[6]	Evaluation	Y	P	P	N	2.0
[7]	Case study	P	Y	P	P	2.5
[8]	SLR	Y	Y	Y	P	3.5
[9]	Evaluation	Y	Y	P	Y	3.5
[10]	SLR	Y	Y	Y	N	3.0
[11]	Review	P	N	N	Y	1.5
[12]	Review	Y	Y	Y	P	3.5
[13]	Evaluation	Y	Y	Y	N	3.0
[14]	Evaluation	Y	P	Y	N	2.5
[15]	SLR	P	Y	Y	P	3.0
[16]	Evaluation	Y	Y	P	P	3.0
[17]	Evaluation	Y	P	P	P	2.5
[18]	Survey	Y	N	N	P	1.5
[19]	Case study	Y	Y	Y	Y	4.0
[20]	Case study	Y	Y	Y	P	3.5
[21]	Case study	Y	Y	Y	N	3.0

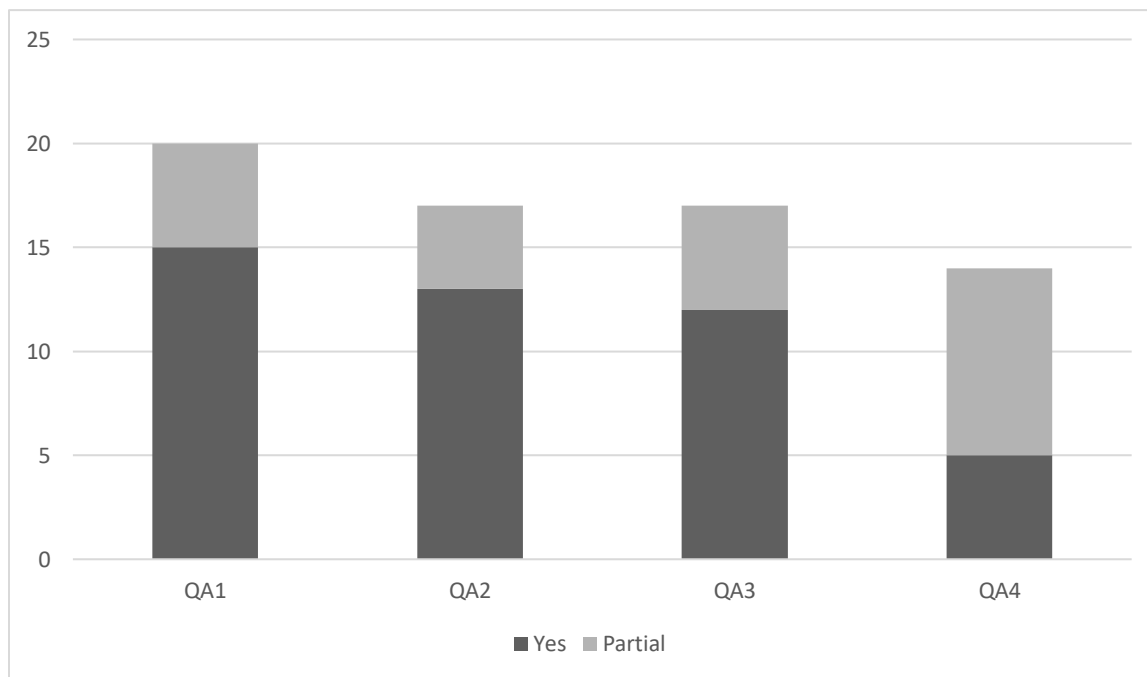


Figure 2: Quality assessment coverage distribution

A. What research topics are being addressed?

All the research papers selected for this literature review are related to the topic of documentation in requirements engineering.

Ten [2, 3, 7, 8, 10, 12, 13, 15, 18, 19] out of the twenty selected papers contain discussions on requirements documentation in agile software teams and current practices.

There are fourteen [2, 4, 7, 8, 10, 11, 13, 14, 15, 16, 17, 18, 19, 20] out of the twenty selected papers discussed solution to the challenges to some extent.

B. What are the trends for the research methodology?

All seven evaluation papers [3, 5, 8, 12, 13, 15, 16] utilized qualitative and quantitative evaluation methods to assist in their analysis of respective studies.

There are four SLR [2, 7, 9, 14] in the list of selected papers which utilizes a set of selection criteria. The list of selection criteria is built from the research questions that are established, and a quality assessment is carried out to determine if the searched paper is selected.

In the remaining case study papers, the authors made selection on a few major industry and domain to conduct the research of respective research areas.

C. What are the current practices of documenting NFRs in agile?

In most software projects, NFRs are communicated by the stakeholders during the early stages of the project. They are usually documented in minutes of meetings (MOMs), formally documented in WIKI systems or translated to user stories [19] to be delivered by the development teams.

NFRs do contain quality elements of the software system, such as performance, availability and security [4], among others. These are the attributes that should be present in the system at any given point in time.

In most cases in agile approaches, these NFRs are seldom revisited as development sprints iterate, until a point where quality dips beyond a tolerable threshold that a bug or defect is reported. ASD teams will then revisit the expectation as a new user story in order to improve the quality back to the expected level. This practice is reactive in nature and it can be costly to remedy.

There are also new functional requirements that carry specific NFRs, which are delivered during the development sprint [8]. However, these ad-hoc NFRs are not consistent with the overall quality expectations of a given software system architecture.

D. What are the challenges of NFRs documentation?

There are two main challenges in NFRs documentation: validity and management [2, 7, 19, 20].

As mentioned in the previous section, NFRs apply to a software system in general and are usually communicated at the start of any given software projects. These NFRs get outdated when new expectations are communicated by the stakeholders. As a result, earlier documents which are not updated may cause conflicts and confusion [20], when the new user story is not linked and reference correctly to maintain requirements traceability.

When NFRs are communicated by stakeholders, they are usually not expressed in measurable technical terms that can be easily understood by the programmer [2, 6]. Often, documentations contain vague or unreasonable details that could cause confusion or undeliverable requirement when translated into a user story.

In agile, requirements are documented on a high level as user story. The details of the requirements are usually communicated by the stakeholders through collaboration [4]. It takes time and effort to ensure that the requirement

details are accurately documented. Therefore, it is common to find that teams do not spend the time to document the expectations accurately [16, 19], instead time is spent to implement and ensure the shortest time to market instead. Tracing back these requirements can be very challenging especially when team composition turnover is high [4].

E. What are the proposed solutions?

For resourceful teams in terms of available headcounts and funding, the role of eliciting proper requirements and documentation can be delegated to personnel that focus to ensure the accuracy and traceability of NFRs in the software project [3]. In big organisations, these roles include technical product owners, business and system analysts and so on. The responsibilities of documenting requirements should not be given to the engineering team because in most cases will not put documentation and requirements as a priority [17].

During the process of grooming every user story, the impact and risk towards all NFRs shall be analysed and properly linked to the user story itself, so that developers working on the user stories are aware of the non-functional expectations and impacts. Modern software development planning tools [7, 18, 20] such as TFS and JIRA can be utilized to achieve this outcome.

ASD teams can also ensure the NFR documentation is always up to date by keeping customized NFRs to a minimum. NFRs that are common and can be applied to most software systems can utilise common standards and best practices by the software development industry. Doing so can keep the effort of maintaining the documentation low, and software quality can be easily achieved by using existing automation tools [19] available in the industry, such as penetration testing suite (security), static code analysis tools (code quality), and so on.

Lastly, it is crucial that the importance of NFR documentation is given the right amount of priority in every organization. The leadership team should provide ample support to engineering teams to ensure the documentation is always up to date. This will eventually cascade and influence all other functions within the organization to ensure that NFR documentations are reliable and correct.

The challenges highlighted in the findings in earlier sections are less impactful to other development manifesto such as waterfall approach. However, these limitations should not be a deterrent to practice agile because the agile manifesto offers many other values that are beneficial to the modern software development climate.

F. What are the limitations presented in the existing researches?

At the end of the literature review of all selected papers, there are four papers [2, 5, 6, 18] which highlighted limitations in their respective researches.

There is a lack of adequate research in the topic of documentation of NFR in software teams practicing agile methodologies [19]. The effects and impacts of NFR documentation has not been given enough attention across the industry.

There is also a certain degree of occupational bias [2, 18] in the respective research when evaluating the effectiveness of delivering NFR. The importance of software quality has reduced when teams moved from traditional requirements gathering into agile practices.

Researches of this nature are often too general when stereotyping and classification of ASD teams in general [2, 18]. Not all organisation culture and software teams can be easily grouped into convenient classes across all researches.

In surveys conducted for case studies, the responses also pose a risk of inaccuracies when respondents do not provide the true representation of reality [3].

Lastly, the result of some research papers used for reference are also outdated because the software development industry is a very fast-moving pace where practices, techniques and management policies change very frequently to adapt to the needs and demands of software delivery.

G. Limitations of this study

This review only focuses on research findings from the last five years, with time constraint and limited review capability. As such, the contents and materials of literature under study contains the highlights that are of attention in recent years. Therefore, it does not provide a comprehensive overview of the requirements documentation of NFRs that affects software teams in a general perspective since usually, new pressing issues are given most of the attention while old problems, though unsolved are slowly neglected over time.

The majority of literatures reviewed focuses in the agile process and project management. The scope covering utilisation of technology or tools that can affect the outcome for software teams practicing agile is very minimal. In recent years, there are many new development tools and systems that target agile software development methodologies that could have provided new approaches to resolve the challenges of NFRs. Some examples of these tools are expanded capabilities of containerization and orchestration, new capabilities on cloud service providers, new monitoring and diagnostics tools, and so on.

As with other SLRs that are used to conduct this review, there are also elements of bias with the assumption that the problems and solutions of requirements documentation in ASD teams can be conveniently resolved by including a few positive changes to the way requirements are delivered, agreed and tracked. These methods are only suitable in the context of huge organisations or specialised software teams, but did not account for software teams in smaller companies or short-term software projects. In that scenario, functional requirements are usually the main focus of project deliverable goals, and quality metrics comes in distant second priority.

V. CONCLUSION

In conclusion, this systematic literature review covers all the research questions that are considered. The importance of documentation for NFR is highlighted, studied and proven to yield positive outcomes.

NFRs elicitation and documentation receives very low priority and importance across most ASD teams. When very few resources are invested in this area, especially in agile environments – the effects of poor NFRs documentation ripple downwards to many aspects of development practices, processes and quality of deliverables.

There are many literatures that study the practices and challenges of requirements elicitation and documentation of NFRs in agile, as well as proposing a few ideas that can be adopted by teams in their existing workflow to improve the situation. Further studies can be looked at from the agile manifesto context to discover significant improvement to the manifesto to provide a better solution to requirements documentation.

ACKNOWLEDGEMENT

The authors received no funding from any party for the research and publication of this article.

REFERENCES

- [1] Kitchenham, A. Barbara, T. Dyba, and M. Jorgensen. "Evidence-based software engineering." Proceedings. 26th International Conference on Software Engineering. IEEE, 2004.
- [2] A. Alhazmi and S. Huang, "Survey on Differences of Requirements Engineering for Traditional and Agile Development Processes," 2020 SoutheastCon, pp. 1-9, 2020.
- [3] A. Jarzębowicz and P. Weichbroth, "A Qualitative Study on Non-Functional Requirements in Agile Software Development," in IEEE Access, vol. 9, pp. 40458-40475, 2021.
- [4] B. Boehm, D. Rosenberg and N. Siegel, "Critical Quality Factors for Rapid, Scalable, Agile Development," 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C), pp. 514-515, 2019.
- [5] B. Habib and R. Romli, "A Systematic Mapping Study on Issues and Importance of Documentation in Agile," 2021 IEEE 12th International Conference on Software Engineering and Service Science (ICSESS), pp. 198-202, 2021.

- [6] C. Werner, Z. S. Li, N. Ernst and D. Damian, "The Lack of Shared Understanding of Non-Functional Requirements in Continuous Software Engineering: Accidental or Essential?", 2020 IEEE 28th International Requirements Engineering Conference (RE), pp. 90-101, 2020.
- [7] X. Franch et al., "Data-Driven Elicitation, Assessment and Documentation of Quality Requirements in Agile Software Development", Krogstie, J., Reijers, H. (eds) *Advanced Information Systems Engineering. CAiSE. Lecture Notes in Computer Science*, vol 10816. Springer, Cham, 2018.
- [8] P. Heck., A. Zaidman, "A systematic literature review on quality criteria for agile requirements specifications", *Software Qual J* 26, pp. 127–160, 2018.
- [9] J. Pasuksmit, P. Thongtanunam and S. Karunasekera, "Towards Just-Enough Documentation for Agile Effort Estimation: What Information Should Be Documented?," 2021 IEEE International Conference on Software Maintenance and Evolution (ICSME), pp. 114-125, 2021.
- [10] J. Coutinho, L. Wilkerson, Andrade, and P. Machado., "Requirements Engineering and Software Testing in Agile Methodologies: a Systematic Mapping. In *Proceedings of the XXXIII Brazilian Symposium on Software Engineering*", (SBES 2019). Association for Computing Machinery, New York, NY, USA, pp. 322–331, 2019.
- [11] K. A. Memon, X. Xia, and H. Halepoto. "Some Serious Deliberations and Reflections on the Significance of Non-Functional Requirements for Improving the Quality of Software", 2019 8th International Conference on Software and Information Engineering (ICSIE '19). Association for Computing Machinery, New York, NY, USA, pp. 38–41, 2019.
- [12] M. Shafiq and U. s. Waheed, "Documentation in Agile Development A Comparative Analysis," 2018 IEEE 21st International Multi-Topic Conference (INMIC), pp. 1-8, 2018.
- [13] M. Younas et al., "Elicitation of Nonfunctional Requirements in Agile Development Using Cloud Computing Environment," *IEEE Access*, vol. 8, pp. 209153-209162, 2020.
- [14] M. Amorndettawin and T. Senivongse., "Non-functional Requirement Patterns for Agile Software Development", 2019 3rd International Conference on Software and e-Business (ICSEB 2019). Association for Computing Machinery, New York, NY, USA, pp. 66–74, 2019.
- [15] Nurbojatmiko, K. E. Budiardjo, and W. C. Wibowo., "Slr on Identification & Classification of Non-Functional Requirements Attributes, and Its Representation in Functional Requirements", 2018 2nd International Conference on Computer Science and Artificial Intelligence (CSAI '18). Association for Computing Machinery, New York, NY, USA, pp. 151–157, 2018.
- [16] M. Oriol, S. Martínez-Fernández, W. Behutiye et al., "Data-driven and tool-supported elicitation of quality requirements in agile companies", *Software Qual J* 28, pp. 931–963, 2020.
- [17] G. Robiolo, E. Scott, S. Matalonga, M. Felderer, "Technical Debt and Waste in Non-functional Requirements Documentation: An Exploratory Study", *Product-Focused Software Process Improvement. PROFES 2019. Lecture Notes in Computer Science*, vol 11915. Springer, Cham., 2019.
- [18] S. Tariq and S. M. Cheema, "Approaches for Non-Functional Requirement Modeling: A Literature Survey," 2021 4th International Conference on Computing & Information Sciences (ICIS), pp. 1-6, 2021.
- [19] W. Alsaqaf, M. Daneva and R. Wieringa, "Understanding Challenging Situations in Agile Quality Requirements Engineering and Their Solution strategies: Insights from a Case Study," 2018 IEEE 26th International Requirements Engineering Conference (RE), pp. 274-285, 2018.
- [20] W. Behutiye, P. Rodríguez, M. Oivo, S. Aaramaa, J. Partanen and A. Abhervé, "How agile software development practitioners perceive the need for documenting quality requirements: a multiple case study," 2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), pp. 93-100, 2020.
- [21] W. Behutiye, P. Seppänen, P. Rodríguez, and M. Oivo. "Documentation of Quality Requirements in Agile Software Development", *Proceedings of the Evaluation and Assessment in Software Engineering (EASE '20)*. Association for Computing Machinery, New York, NY, USA, pp. 250–259, 2020.