A Systematic Literature Review on Requirements Engineering Practices and Challenges in Open-Source Projects

Maliha Tasnim, Maruf Rayhan, Zheying Zhang, Timo Poranen
Tampere University
Tampere, Finland
firstname.lastname@tuni.fi

Abstract—**Open-source software (OSS) development has become increasingly influential in the software industry, promoting collaboration and knowledge sharing among developers and users. Along with rapidly evolving OSS projects, this paper explores requirements engineering (RE) practices and challenges through a systematic literature review (SLR). Synthesizing data from 43 selected papers, the study reports practices, techniques, and methods that assist RE activities in OSS projects, and also addresses challenges faced by practitioners and the potential solutions. The results of the literature review indicate a growing interest in using machine learning and statistical methods to assist RE activities, focusing on automated requirements identification and analysis using information from project discussion forums, issue reports, and other online resources. The findings also highlight the importance of community involvement, with many studies examining developers' interaction patterns, expertise levels, and influence on projects. These findings provide valuable insights for OSS project managers and researchers, offering guidance on effectively handling requirements in OSS projects.**

Index Terms—**requirements engineering, open-source software development, systematic literature review (SLR)**

I. Introduction

Requirements engineering (RE) is the process of eliciting, analyzing, documenting, validating, and managing the needs and expectations of stakeholders, ensuring that the software product meets their needs [3] [14] [16]. Establishing a shared understanding of the requirements among stakeholders is essential for successfully developing, maintaining, and evolving a software product, with various techniques and tools to support the communication and cooperative interaction process between requirements analysts and stakeholders [13] [6].

Open-source software (OSS) projects are characterized by a distributed community of developers, users, and other stakeholders, working in a collaborative and often decentralized manner. Given the less formal organizational structure in OSS projects, the RE process diverges from conventional software development practices. Requirements are typically expressed for open review, elaboration, and discussion through informal channels such as mailing lists, forums, and issue tracking systems [15] [11]. Traditional requirements review activities are replaced by community feedback and testing. Comments and peer reviews are common practices in OSS projects, where developers assess each other's work to identify and resolve potential issues and discrepancies. By continuously building, integrating, and testing change requests and implementations, developers can ensure that requirements are properly implemented and satisfied [1]. This results in more dynamic and informal RE activities.

Given these observations, this work aims to explore the requirement management practices and challenges in the OSS context through a systematic literature review (SLR). The purpose of this paper is 1) to identify RE practices to support OSS project managers and contributors in requirements creation, analysis, and management, thus enhancing their effectiveness and efficiency in conducting requirements analysis and implementation in OSS projects; and 2) to recognize challenges and research-based solutions for these RE activities in OSS projects. To achieve these goals, we formulated two research questions (RQs).

RQ1 What are the requirements engineering practices in open-source projects according to published studies?

RQ2 What are the challenges and proposed solutions associated with requirements engineering practices in open-source projects?

The remainder of the paper is structured as follows. Section II covers existing literature reviews on OSS development, and identifies a gap in the literature and a need for investigating RE practices in OSS projects. Section III presents our research method in detail, and Section IV summarizes the key findings of our study. Section V further interprets the main findings and section VI discusses the threats to the validity of the research. Section VII concludes the work.

II. Related Work

A number of systematic literature studies and mapping studies have been conducted to explore various aspects of OSS development, ranging from investigating the distinctions between traditional and OSS development activities to assessing community participation and engagement. A summary of the related work is given in Table I. Castro Llanos and Acuña Castillo [2] identified 22 primary studies in their systematic mapping study and reviewed software development processes enacted by the OSS community, with a focus on the requirements, design, and implementation activities in OSS projects. Gandomani et al. [5] evaluated the relationship between agile software development (ASD) and OSS development methods, finding evidence that agile and open-source practices complement each other. The findings showed that incorporating some ASD practices in OSS development was feasible. Kiran and Ali [8] explored requirement elicitation techniques in OSS projects, inspecting how the requirement elicitation process is carried out through a literature survey. Furthermore, Franco-Bedoya et al. [4] assessed the current state of the art in OSS ecosystems (OSSECOs) research through a mapping study of 87 primary studies. The aim of this research was to explore and understand the genesis of the OSSECO terms from related definitions, the software ecosystems in the context of OSS, as well as the modeling and analysis techniques of OSSECOs. In a more recent study, Kaur et al. [7] conducted a systematic mapping study to gain a comprehensive understanding of community participation and engagement in OSS projects. Their findings provide insights into the dynamics of OSS projects and highlight the significance of community contributions and factors that drive and influence community engagement.

While the related work explored various aspects of OSS development, none of the above-mentioned studies comprehensively investigated RE practice in OSS projects. An SLR focusing on RE activities would fill this gap, providing a complete understanding of the challenges and practices associated with requirements analysis and management in OSS projects. Our research aims to identify the solutions and practices for OSS project managers and contributors involved in requirement identification, analysis, and management, as well as examine the latest research on requirements practices in OSS projects and potential challenges and solutions.

TABLE I
RELATED WORK

Studies	Goals	Research Questions	Research method
Llanos and Castillo, 2012 [2]	Review of software development processes enacted by the OSS community	What activities do OSS process models contain?	Mapping study
Gandomani et al. 2013 [5]	Assessment of the relation and integration of agile software development (ASD) and open source software development methodology (OSSD)	RQ1: Could ASD and OSSD have any relationship? RQ2: Are practices of one of them applicable in the second? RQ3: Can they integrate with each other?	Literature review
Franco-Bedoya et al. 2017 [4]	Evaluation of the current state of the art in OSS ecosystems (OSSECOs) research	RQ1. What are the demographic characteristics of the studies about OSSECOs? RQ2. What is an OSSECO? RQ3. Which representations have been proposed for OSSECOs?	Mapping study
Kiran and Ali, 2018 [8]	Investigation of requirements elicitation techniques in open source projects	How the process of requirement elicitation is carried out for open source software?	Literature survey
Kaur et al. 2020 [7]	Review of the community participation and engagement in OSS projects	RQ1: Which studies have been published in the literature related to community dynamics? RQ2: What empirical evidence is provided for the topics addressed in the community dynamics studies?	Mapping study

III. RESEARCH METHOD

In this section we describe literature search strategy and data extraction from the selected studies. We adopted comprehensive guidelines by Kitchenham and others [9], [10] to establish our search strategies and the review protocol before conducting the systematic literature review.

A. Search Strategy

1) Search String: (”open source software” OR oss) AND (project OR development)) AND (”requirements management” OR ”requirements engineering” OR ”requirements analysis” OR ”handle requirements”) AND (process OR tool OR technique OR activit* OR challeng* OR solution OR benefit))

The search string is formulated iteratively, incorporating keywords identified from research questions and diversified using synonyms. We start with the terms ”requirements management”, ”open source software project”, ”process”, ”tool”, and ”challenge”, and continue to identify the synonyms, abbreviations, or alternatives to those words or terms to widen the coverage of the search. This is also the reason for using Asterix (*) to include variations of certain keywords in the search. We concatenate the key terms using AND so that we can search for relevant studies, and concatenate the alternatives and synonyms using OR so that we can maximize the percentage of search results being relevant. To ensure consistency and completeness, we verify the presence of terms in several iterations to look for relevant terms from papers chosen from the rounds of search results.

2) Information sources and search process: The same search string is used to search for relevant studies. The study focuses on four electronic sources known for their relevance to software engineering: Scopus, IEEEXplore, ACM Digital Library, and Web of Science. Scopus, the largest database of abstracts and citations [9], was used as the initial database for the search. Duplicate records found in Scopus were excluded from the research results obtained from other databases. Overall, 256 unique results came from database searches.

3) Inclusion and exclusion criteria: The inclusion and exclusion criteria for our study have been specified based on our research questions. The inclusion criteria require that studies investigate relevant topics to address our research objectives. We include studies that meet four specific criteria. Exclusion criteria include those commonly used in SLR, such as non-English papers, duplicate papers, non-peer-reviewed papers, research plans and roadmaps, and the use of a secondary study method. Additionally, studies related to OSS adoption, evaluation of a requirements tool using OSS projects, or the development of requirements tools are outside the scope of this study and are listed in the exclusion criteria. The complete list of inclusion and exclusion criteria is given in Table II.

The initial screening of search results for relevant studies was conducted independently by the first two authors of this paper, who applied the inclusion and exclusion criteria to select relevant papers. The results were compared after the independent screening process to reach a consensus on the

TABLE II
INCLUSION AND EXCLUSION CRITERIA

Incl./Excl.	Criteria
Inclusion	Papers on open-source software development process
	Papers on how to handle issues and contributions in OSS projects
	Papers applying methods, techniques, or tools for requirements elicitation, analysis and management in OSS projects
	Papers analyzing contributors' influence in handling requirements for OSS projects
Exclusion	Not in English
	Focuses on OSS evaluation and adoption
	Focuses on requirements specification for an OSS project
	Focuses on using OSS projects as examples for tool/method evaluation
	focuses on open-source tools
	Duplicate papers
	Out of topic and using the terms for other purposes
	Non-peer-reviewed papers
	Secondary studies, vision papers, or tutorial

papers to be included. If there is a dispute between results, the first two authors would read the full paper, discuss it, and then confirm with the third author whether the paper should be included or not. In addition, the selected papers underwent assessment for reporting quality and study generalizability, with short reports being excluded from the final selection. In total, 39 papers were included in the study.

Each of the 39 papers underwent forward and backward snowballing in parallel [17]. We reviewed all the references listed in the selected papers and evaluated all the papers that reference the selected ones. The process provided 3 additional papers in forward snowballing and 1 additional in backward snowballing. Hence, the total number of selected papers was 43. The number of papers resulting from each process is summarised in Fig 1.

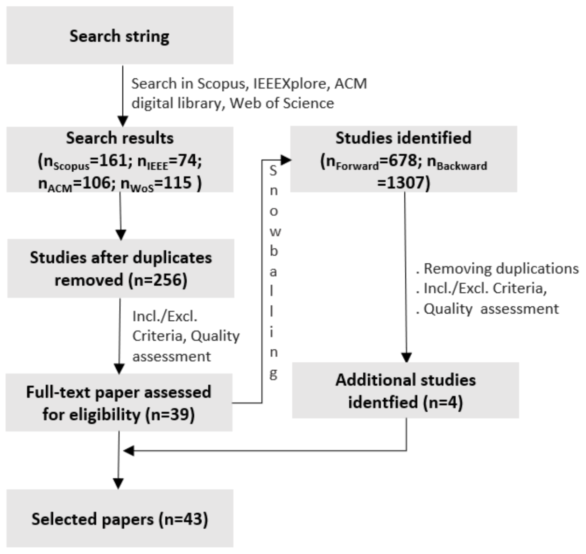

Search string
Search in Scopus, IEEEXplore, ACM digital library, Web of Science
Search results (n_{Scopus}=161; n_{IEEE}=74; n_{ACM}=106; n_{WoS}=115)
Studies after duplicates removed (n=256)
Incl./Excl. Criteria, Quality assessment
Full-text paper assessed for eligibility (n=39)
Snowballing
Studies identified ($n_{Forward}$=678; $n_{Backward}$ =1307)
. Removing duplications
. Incl./Excl. Criteria,
. Quality assessment
Additional studies identfied (n=4)
Selected papers (n=43)

Fig. 1. Search and selection process

B. Data Extraction

In order to extract data from the 43 studies, we developed a coding schema based on our research objectives and questions. Our coding process was conducted in three phases. Firstly, we selected eight papers from the list and had the first two authors extract data independently using the coding schema. We then compared the extracted data, revised the data items, and refined the coding schema. Our coding schema encompasses various characteristics extracted from the selected studies, including their goals, research methods, requirements management activities and practices reported, methods and tools used, challenges faced in requirements management in OSS development, and proposed solutions. After agreeing on the coding schema, the first two authors coded the papers using the schema in the second stage. We then held discussion meetings to interpret and analyze the codes based on perceived similarities, and to discuss any confusion and discrepancies. In cases of disagreement, the full paper was read again by the first two authors and the last two authors were involved in the discussion and the decision-making process. In the third phase, we summarized the data by identifying themes that emerged from the codes. These identified themes formed the categories reported in the Results section.

IV. RESULTS

We selected 43 papers through the process described earlier. The papers are listed in Appendix A and cited as SP* in the following discussion. They were published from 2002 to 2022. The distribution of the number of papers over time is shown in Fig 2. Among these 33 papers were published in conferences and 10 were published in journals.

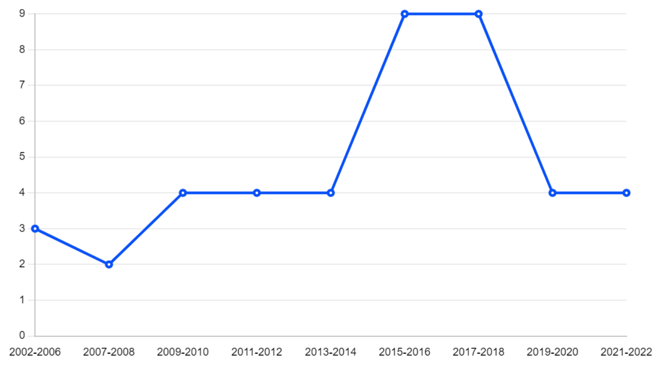
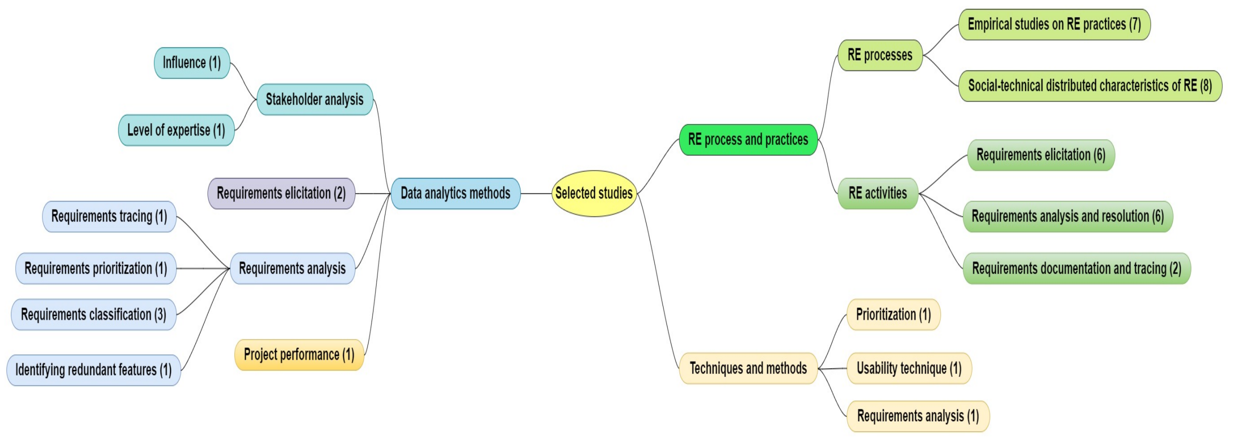


Fig. 2. Year wise distribution of selected studies

The selected studies were analyzed and grouped into two categories based on their research topics inferred from the extracted data, as shown in Fig. 3 Out of the total 43 studies, 29 (67.44%) specifically address the characteristics of the requirements engineering process and practices in OSS projects. The remaining 14 studies (32.56%) investigate various techniques and methods that support requirements-related tasks, with 11 studies (25.58%) notably focusing on data analytics methods for analysis.

The 29 studies on RE processes and practices revealed several main themes. Numerous studies focused on the social-technical distributed aspects of RE practice in open source communities, examining the processes and frameworks involved [SP9], [SP10], [SP15], [SP16], [SP18], [SP25], [SP34],



Selected studies
Data analytics methods
Stakeholder analysis
Influence (1)
Level of expertise (1)
Requirements elicitation (2)
Requirements analysis
Requirements tracing (1)
Requirements prioritization (1)
Requirements classification (3)
Identifying redundant features (1)
Project performance (1)
RE process and practices
RE processes
Empirical studies on RE practices (7)
Social-technical distributed characteristics of RE (8)
RE activities
Requirements elicitation (6)
Requirements analysis and resolution (6)
Requirements documentation and tracing (2)
Techniques and methods
Prioritization (1)
Usability technique (1)
Requirements analysis (1)

Fig. 3. Study topics divided into categories

and [SP40]. Meanwhile, other research investigated the informal and open innovation nature of software engineering practices, particularly regarding the RE process [SP13], [SP24], [SP30], [SP42], [SP8], [SP21], and [SP26]. Furthermore, several papers conducted an empirical study on specific RE activities, e.g. requirements elicitation [SP14], [SP17], [SP20], [SP23], [SP32], [SP39], analysis [SP5], [SP12], [SP19], [SP31], [SP36], [SP43], and documentation and tracing [SP35] and [SP38]. These studies explored various aspects of RE in the context of OSS development.

Out of the 14 studies on techniques and tools for requirements analysis and management, 11 recent studies proposed and evaluated data analysis methods such as natural language processing (NLP), machine learning (ML), or statistical analysis algorithms to accomplish various tasks. These tasks include analyzing stakeholders' influence [SP3] and expertise [SP29], eliciting requirements from developer communities or forums [SP6] [SP28], identifying redundant features [SP33], classifying or prioritizing requirements [SP1], [SP22], [SP27], [SP4], tracing requirements [SP37] and monitoring project performance [SP2]. Additionally, 3 studies proposed and evaluated requirements elicitation and analysis techniques that primarily relied on participants' discussions and voting [SP4], [SP7], and [SP11].

A. RQ1 What are the requirements engineering practices in OSS projects according to published studies?

Synthesizing data from 43 selected papers, we summarize the practices, techniques, and methods that support RE activities in OSS projects in Table III.

TABLE III
REQUIREMENT ENGINEERING PRACTICES

Requirement Engineering Activities	**Practices**	**Related Papers**
Requirements elicitation	Assertion & discussion in developer's community	SP8, SP14, SP17, SP23, SP25, SP30, SP31, SP32, SP34, SP40, SP42
	Automated requirements identification	SP27, SP28, SP43
Requirements analysis	Automated requirements classification	SP1, SP4, SP6, SP33, SP43
	Requirements prioritization	SP7, SP8, SP14
	Refinement and resolution	SP5, SP12, SP14, SP19, SP25
	Requirements analysis	SP25, SP31, SP32, SP34, SP35, SP36, SP40, SP42
Requirements documentation	Informal documentation	SP14, SP25, SP34, SP35, SP38, SP40, SP42
Requirements management	Feature management	SP32
	Traceability	SP26, SP31, SP32, SP35, SP37, SP40, SP38
Stakeholder analysis	Community involvement	SP17, SP39, SP42
	Data analytics methods	SP2, SP3, SP18, SP20, SP28, SP29
Social-technical characteristics	Socio-technical distributed cognitive process	SP9, SP10, SP15, SP16
	Informal requirements	SP25, SP30
Others		SP11, SP13, SP21, SP22, SP24, SP41

1) Requirements elicitation: Proposing and identifying change requests and new features that align with a project's roadmap is crucial in software development and ongoing improvements. Numerous studies have emphasized the responsibility of core developers and team members in determining the majority of features (SP8, SP14, SP23, SP25, SP32, SP34, SP42). Feature requests and bug reports are often conveyed through provisionments, a statement that describes a feature provided by existing software systems, competing products, or prototypes put forth by developers advocating for the changes(SP30). These provisionments and assertions are openly discussed within the developer community, drawing upon personal experiences or knowledge of user needs. Such discussions typically take place on forums, during project workshops, through emails, or by using reporting tools such as Jira, Bugzilla, or other issue tracking systems (SP8, SP17, SP25, SP32, SP34). Meanwhile, studies SP40 and SP23 have suggested improvements to the decision-making process for selecting proposed use cases, constraints, and feature requests. Furthermore,to automate the labor-intensive process of identifying requirements from diverse information sources, NLP techniques, ML methods, and grammar-based parsing strategies (SP43) have been applied in recent studies. For example, a logistic regression model has been proposed to identify security requirements from reported issues (SP27).

2) Requirements analysis and validation: Requirements analysis involves tasks such as identifying redundant requirements and classifying and prioritizing issues and change requests. These tasks rely on automated methods, contributors' feedback and comments, and diverse information sources. Several studies proposed automated approaches for requirements classification, using and comparing algorithms such as TF-IDF, Naive Bayes, Decision Tree, and Random Forest (SP1, SP4).

Additional automated approaches have been suggested for identifying redundant feature requests (SP33) and classifying security requirements (SP43). Besides automated methods, requirements prioritization and refinement often involve informal discussions, comments, and voting processes, where community members, e.g. project maintainers, developers, and users, collaboratively assess the proposed features and improvements (SP5, SP8, SP14, SP25). Requirements validation is typically conducted through unit testing and integration testing of change requests (SP7, SP8, SP14, SP19). Researchers have explored community communication, scenarios of usage, HowTo guides, external publications, bug reports, issue tracking, and documentation as sources for requirements analysis (SP14, SP25). Furthermore, several studies have explored approaches and frameworks to better understand OSS requirements and address communication problems in OSS development (SP34, SP36, SP40).

3) Requirements documentation: Documentation is typically less formalized in OSS projects. Studies indicate that requirements are documented through discussion and voting (SP14, SP25, SP34, SP40, SP42) and in various formats such as forums, guidelines, READMEs, wikis, and website resource files (SP35, SP38). Although not always required, templates for reporting issues and generating change requests are frequently used in OSS projects to provide guidelines for contributors when submitting their work or reporting problems. For instance, guidelines for documenting and managing issues for agile software development on Github have been proposed in studies (e.g. SP35).

4) Requirements management: In OSS projects, requirements management and tracing are achieved through issue tracking systems, source control systems, informal documentation, and tags or labels. The degree of traceability varies based on project size, complexity, community involvement, as well as the abstraction level of requirements. Concrete and technical requirements are often reported in the project's issue tracker and can be traced with the support of issue references and status tracking (SP35, SP40), while features documented in READMEs, wikis, and website resource files may lack explicit referencing to source code (SP38). Additionally, Requirements tracing has been explored through the use of automated approaches based on the Universal Sentence Encoder following a semantic search and innovative clustering technique (SP26), Vector Space Model (VSM), Term Frequency-Inverse Document Frequency (TF-IDF) techniques (SP37).

5) Stakeholder analysis: Many studies highlight the importance of community involvement and participation in OSS projects, as well as the need for effective stakeholder management and communication throughout the software development life cycle. These studies result in preliminary guidelines for the organization of community-oriented software development (SP17, SP39, SP42). Many recent studies applied quantitative analysis techniques such as social network analysis (SP2, SP3, SP18, SP20), Markov Network (SP28, SP29), and graph theory (SP20) to model and analyze stakeholder interactions patterns and their influence, and their expertise level to a given topic.

6) Social-Technical Characteristics: OSS projects represent complex relationships between the social aspects (e.g. collaboration, communication, stakeholders) and technical aspects (e.g. software architecture, code base, tools). Understanding these characteristics is essential for promoting a healthy OSS ecosystem. Some studies have investigated the impact of social-technical characteristics on OSS projects, examining the role of distributed and cognitive activities across releases (SP9, SP10, SP15) and the influence of internal social capital in driving RE activities (SP16). Studies highlight that informal requirements are common in OSS projects (SP25, SP30).

Moreover, various other aspects of RE practices in OSS projects have been explored, including the impact of crowdsourcing on OSS project performance and process effectiveness evaluation (SP2), the feasibility of adopting the visual brainstorming usability technique in OSS project (SP11), models for analyzing and managing requirements in OSS and Software Ecosystems (SECOs) (SP13), challenges in managing requirements in an open innovation context (SP21), as well as the frameworks and models to describe the requirements analysis process in OSS projects (SP24, SP41).

B. RQ2 What are the challenges and proposed solutions associated with requirements engineering practices in open-source projects?

Studies have demonstrated that diverse challenges may arise in the process of OSS development. The challenges can be grouped into two distinct categories, with various studies exploring and proposing solutions to address them.

1) Managing the large volume of issues and feature requests: Several studies have highlighted the challenges in managing the growing volume of system requirements, feature requests, and issues in OSS projects (SP2, SP15, SP21, SP22, SP32). There is often a lack of support for handling and managing a large number of feature requests in forums and wikis, making it harder for users to submit well-thought-out requests (SP33) and for developers to visualize and navigate extensive networks of feature requests (SP37). These challenges affect the overall RE process by making it difficult to understand, identify, classify, and clarify requirements, leading to delays in software development, increased bugs and errors, and higher development iterations (SP36).

To address these challenges, various solutions have been proposed. These include the implementation of processes to organize and prioritize issues and change requests, the application of CrowdRE techniques, and the assignment of dedicated managers to effectively manage input from the crowd (SP2). Utility-based prioritization techniques have been proposed to facilitate decision-making in selecting the most important and urgent requirements and assigning them to suitable developers for review and implementation (SP7). Furthermore, recent research trends involve utilizing ML methods to handle the large volume of requirements, such as automated requirements identification and analysis (SP1, SP4, SP6, SP27, SP28, SP33, SP43). For instance, a prioritization tool was proposed in SP4 to recommend relevant requirements (issues/bugs) to open-source developers. The study used data from Eclipse to build a prediction model, training and evaluating different classifiers. The results revealed that the Random Forest classifier offered the highest precision.

2) Distributed nature of OSS projects: The distributed nature of OSS projects presents unique challenges in communication, management of requirements, and resource allocation due to the cognitive distribution among multiple developers and their interaction in social networks. RE is considered to be less formal and dependent on online documentation and communication tools. To address these challenges, several approaches have been proposed, including stakeholder influence analysis (SIA) methods that analyze developers' social networks and evaluate their influence and expertise (SP2, SP3, SP18, SP20, SP28, SP29) to identify responsible developers for requirements analysis and implementation. In particular, the study in SP20 investigated the impact of structural hole theory on the identification of new requirements in OSS projects. The findings enhance the RE practice in decentralized environments by understanding the influence of social network structures on OSS development.

Challenges like resource constraints, conflicting stakeholder interests, and coordination issues (SP12) can lead to delays in elaborating, refining, and resolving just-in-time requirements. To address these challenges, solutions have been proposed including enhancing communication and coordination among contributors, appropriate resource allocation, prioritizing requirements based on stakeholder feedback, and resolving stakeholder conflicts through negotiation and mutual compromise (SP12).

In addition, solutions have been proposed to tackle these challenges, for instance utilizing a universal sentence encoder, which helps to identify matched requests coming from forum postings and issue tracker records (SP26), a tool-supported method combining NLP techniques, ML, statistical, and search-based techniques to address the challenge of requirements elicitation from big online discussions (SP28), and a novel framework to handle communication problems and ambiguities in OSSD by merging several approaches (SP36). The studies also recommend the implementation of planning and communication strategies to enhance the decision-making process (SP40) and the need for future research to derive more practical guidelines on the requirements definition for ERP development (SP39).

V. Discussions

In this study, we performed a comprehensive and detailed analysis of RE activities in OSS projects. We synthesized the results and data from numerous studies to analyze the practices, techniques, tools, and methods applied in various RE activities, for instance, requirements analysis, requirements elicitation, documentation, management, stakeholder analysis, and social-technical characteristics. Throughout our findings, the diversified range of RE practices and corresponding tools and techniques for example, NLP techniques, statistical and search-based techniques, ML approaches, etc. applied in OSS projects has been emphasized.

The significance of community involvement and participation in OSS projects and as well as the importance of informal requirements for OSS project development and management are highlighted in our findings, with studies pointing out the importance of effective stakeholder management and communication throughout the software development life cycle. Studies such as SP9, SP10, SP15, and SP16 have analyzed the social-technical characteristics of OSS projects with in-depth reasoning of the complex relationship between social aspects and technical aspects.

To efficiently manage requirements in OSS projects, traceability between requirements and implementation artifacts is vital. Studies suggest that forums are generally used for feature management, while traceability has been implied through automated approaches, recommender systems, and lightweight representations.

Over the past years, the research in RE practices in the context of OSS projects has evolved. Prior to 2010, studies solely drew attention to understanding the social processes, developer community, and informal text descriptions of requirements. Several case studies and surveys were applied to point out how requirements analysis and management differ from traditional RE practices. Since 2009, there has been a deviation in discussing methods and techniques for individual RE activities such as requirements elicitation, prioritization, and tracing. Researchers have shown a growing interest in community involvement and participation in RE development for OSS projects. From then, the use of mixed-methods approaches, combining qualitative and quantitative data techniques was also widely used. The increasing application of NLP and ML methods has led to attention to automotive requirements identification and analysis in OSS projects since 2015, resulting in a surge of publications exploring this topic.

Overall, the findings reported in the selected papers are inspiring as they show an increasing interest in improving RE practices in OSS projects, with a particular focus on community involvement and automation of tasks like requirements identification, classification, and tracing. Looking ahead, we anticipate a strong emphasis on integrating the natural language processing capabilities of large language models (LLMs) and AI tools in requirements analysis. Research trends will likely continue to prioritize community involvement, task automation, and the seamless integration of LLMs and AI tools to enhance the RE process in OSS development.

VI. Threats to validity

To ensure the validity of our SLR study, we established search strategies and a review protocol based on guidelines [9] [12] [10] and followed the four categories recommended in [18] to discuss potential threats. Our search string included keywords identified from research questions and was diversified using synonyms. We executed the queries in major citation databases, including Scopus and Web of Science, and other two major publishers' digital library portals, i.e. IEEE

Xplore and ACM Digital Library, and used snowballing as a complementary method to extend the coverage of studies. We reviewed all the references listed in the selected studies and evaluated all the papers that referenced the selected ones, resulting in 4 additional relevant publications. To reduce the threat to construct validity, the search strategy, review protocol, and data extraction process were entirely based on established guidelines, and study selection and data extraction were conducted independently by the first two authors. To minimize the threat associated with inaccurate data extraction and conclusion validity, we discussed sample extractions to achieve consistency and recorded data in a shared excel file. Discrepancies were discussed and resolved by consensus among all authors. While there is a potential risk of omitting relevant studies, we are confident that our study represents a comprehensive overview of relevant studies within our scope, and any minor threats to external validity are minimal.

VII. Conclusion

In this study, we conducted a systematic literature review on studies addressing RE practices and research in OSS projects. We synthesized the findings from 43 selected studies, identifying the practices, techniques, and methods employed in requirements elicitation, analysis, documentation, management, stakeholder analysis, etc. Our review reports practices and techniques applied in OSS projects, including the use of ML methods, NLP techniques, and other statistical methods for requirements identification and analysis on the basis of diverse sources of information associated with the OSS projects. Interestingly, recent studies indicate a growing potential and interest among researchers and practitioners in integrating AI tools to assist RE activities in practice. Furthermore, our findings emphasize community involvement and participation in OSS projects, highlighting stakeholder dynamics and communication throughout the software development lifecycle.

In summary, the results of this study provide a comprehensive review of the proposed solutions to RE activities in OSS projects. These findings serve as a valuable resource for software practitioners and researchers in the OSS community, assisting them to understand the wide range of practices and techniques utilized in RE activities and providing guidance on effectively managing requirements in OSS projects.

References

[1] Beller, M., Bacchelli, A., Zaidman, A., Juergens, E.: Modern code reviews in open-source projects: Which problems do they fix? In: Proceedings of the 11th working conference on mining software repositories. pp. 202–211 (2014)

[2] Castro Llanos, J.W., Acuña Castillo, S.T.: Differences between traditional and open source development activities. In: Product-Focused Software Process Improvement: 13th International Conference, PROFES 2012, Madrid, Spain, June 13-15, 2012 Proceedings 13. pp. 131–144. Springer (2012)

[3] Cheng, B.H., Atlee, J.M.: Research directions in requirements engineering. Future of Software Engineering (FOSE'07) pp. 285–303 (2007)

[4] Franco-Bedoya, O., Ameller, D., Costal, D., Franch, X.: Open source software ecosystems: A systematic mapping. Information and software technology 91, 160–185 (2017)

[5] Gandomani, T.J., Zulzalil, H., Ghani, A.A.A., Sultan, A.B.M.: A systematic literature review on relationship between agile methods and open source software development methodology. arXiv preprint arXiv:1302.2748 (2013)

[6] de Gea, J.M.C., Nicolás, J., Alemán, J.L.F., Toval, A., Ebert, C., Vizcaíno, A.: Requirements engineering tools. IEEE software 28(4), 86–91 (2011)

[7] Kaur, R., Chahal, K.K., Saini, M.: Understanding community participation and engagement in open source software projects: A systematic mapping study. journal of king saud university-computer and information sciences 34(7), 4607–4625 (2022)

[8] Kiran, H.M., Ali, Z.: Requirement elicitation techniques for open source systems: a review. International Journal of Advanced Computer Science and Applications 9(1) (2018)

[9] Kitchenham, B., Charters, S., et al.: Guidelines for performing systematic literature reviews in software engineering. Tech. Rep. EBSE-2007-01, Keele Durham Univ. (2007)

[10] Kitchenham, B.A., Madeyski, L., Budgen, D.: SEGRESS: Software engineering guidelines for reporting secondary studies. IEEE Transactions on Software Engineering 49(3), 1273–1298 (2022)

[11] Paech, B., Reuschenbach, B.: Open source requirements engineering. In: 14th IEEE International Requirements Engineering Conference (RE'06). pp. 257–262. IEEE (2006)

[12] Petersen, K., Vakkalanka, S., Kuzniarz, L.: Guidelines for conducting systematic mapping studies in software engineering: An update. Information and Software Technology 64, 1–18 (2015)

[13] Pohl, K.: The three dimensions of requirements engineering: a framework and its applications. Information systems 19(3), 243–258 (1994)

[14] Pohl, K.: Requirements engineering: fundamentals, principles, and techniques. Springer Publishing Company, Incorporated (2010)

[15] Scacchi, W.: Understanding requirements for open source software. In: Design Requirements Engineering: A Ten-Year Perspective: Design Requirements Workshop, Cleveland, OH, USA, June 3-6, 2007, Revised and Invited Papers. pp. 467–494. Springer (2009)

[16] Wiegers, K., Beatty, J.: Software requirements. Pearson Education (2013)

[17] Wohlin, C., Kalinowski, M., Felizardo, K.R., Mendes, E.: Successful combination of database search and snowballing for identification of primary studies in systematic literature studies. Information and Software Technology 147, 106908 (2022)

[18] Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in software engineering. Springer Science & Business Media (2012)

Appendix A - List of included studies

[SP1] Pérez-Verdejo, J.M., Sánchez-García, J., Ocharán-Hernández, J.O., Mezura-Montes, E., Cortés-Verdín, K.: Requirements and GitHub issues: An automated approach for quality requirements classification. Programming and Computer Software 47, 704–721 (2021)

[SP2] Robinson, M., Sarkani, S., Mazzuchi, T.: Network structure and requirements crowdsourcing for OSS projects. Requirements Engineering 26, 509–534 (2021)

[SP3] Linåker, J., Regnell, B., Damian, D.: A method for analyzing stakeholders' influence on an open source software ecosystem's requirements engineering process. Requirements Engineering 25, 115–130 (2020)

[SP4] Samer, R., Felfernig, A., Stettinger, M.: Towards issue recommendation for open source communities. In: IEEE/WIC/ACM International Conference on Web Intelligence. p. 164–171. WI '19, Association for Computing Machinery, New York, NY, USA (2019)

[SP5] Bhowmik, T., Do, A.Q.: Refinement and resolution of just-in-time requirements in open source software and a closer look into non-functional requirements. Journal of Industrial Information Integration 14, 24–33 (2019)

[SP6] Wang, W., Mahakala, K.R., Gupta, A., Hussein, N., Wang, Y.: A linear classifier based approach for identifying security requirements in open source software development. Journal of Industrial Information Integration 14, 34–40 (2019)

[SP7] Felfernig, A., Stettinger, M., Atas, M., Samer, R., Nerlich, J., Scholz, S., Tiihonen, J., Raatikainen, M.: Towards utility-based prioritization of requirements in open source environments. In: 2018 IEEE 26th International Requirements Engineering Conference (RE). pp. 406–411 (2018)

[SP8] Munir, H., Linåker, J., Wnuk, K., Runeson, P., Regnell, B.: Open innovation using open source tools: a case study at Sony Mobile. Empirical Software Engineering 23, 186–223 (2018)

[SP9] Gopal, D., Lyytinen, K.: Effects of social structures in requirements quality of open source software project development. In: ICIS 2017 Proceedings. pp. 406–411 (2017)

[SP10] Xiao, X., Lindberg, A., Hansen, S., Lyytinen, K.: "Computing" requirements for open source software: A distributed cognitive approach. Journal of the Association for Information Systems 19(12), 1217–1252 (2018)

[SP11] Llerena, L., Rodríguez, N., Castro, J.W., Acuña, S.T.: How to incorporate a usability technique in the open source software development process. In: Pereira, Ó.M. (ed.) The 30th International Conference on Software Engineering and Knowledge Engineering, Hotel Pullman, Redwood City, California, USA, July 1-3, 2018. pp. 182–181. KSI Research Inc. and Knowledge Systems Institute Graduate School (2018)

[SP12] Do, A.Q., Bhowmik, T.: Refinement and resolution of just-in-time requirements in open source software: A case study. In: 2017 IEEE 25th International Requirements Engineering Conference Workshops (REW). pp. 407–410 (2017)

[SP13] Linåker, J., Wnuk, K.: Requirements analysis and management for benefiting openness. In: 2016 IEEE 24th International Requirements Engineering Conference Workshops (REW). IEEE (sep 2016)

[SP14] Kuriakose, J., Parsons, J.: How do open source software (OSS) developers practice and perceive requirements engineering? An empirical study. In: 2015 IEEE Fifth International Workshop on Empirical Requirements Engineering (EmpiRE). pp. 49–56 (2015)

[SP15] Gopal, D., Lindberg, A., Lyytinen, K.: Attributes of open source software requirements – the effect of the external environment and internal social structure. In: 2016 49th Hawaii International Conference on System Sciences (HICSS). pp. 4982–4991 (2016)

[SP16] Gopal, D.: Effect of social networks on requirements engineering in open source projects. In: 22nd Americas Conference on Information Systems, AMCIS 2016, San Diego, CA, USA, August 11-14, 2016. Association for Information Systems (2016)

[SP17] Neulinger, K., Hannemann, A., Klamma, R., Jarke, M.: A longitudinal study of community-oriented open source software development. In: Nurcan, S., Soffer, P., Bajec, M., Eder, J. (eds.) Advanced Information Systems Engineering. pp. 509–523. Springer International Publishing, Cham (2016)

[SP18] Linåker, J., Rempel, P., Regnell, B., Mäder, P.: How firms adapt and interact in open source ecosystems: Analyzing stakeholder influence and collaboration patterns. In: Daneva, M., Pastor, O. (eds.) Requirements Engineering: Foundation for Software Quality. pp. 63–81. Springer International Publishing, Cham (2016)

[SP19] Bhowmik, T., Reddivari, S.: Resolution trend of just-in-time requirements in open source software development. In: 2015 IEEE Workshop on Just-In-Time Requirements Engineering (JITRE). pp. 17–20 (2015)

[SP20] Bhowmik, T., Niu, N., Singhania, P., Wang, W.: On the role of structural holes in requirements identification: An exploratory study on open-source software development. ACM Trans. Manage. Inf. Syst. 6(3) (sep 2015)

[SP21] Wnuk, K., Pfahl, D., Callele, D., Karlsson, E.A.: How can open source software development help requirements management gain the potential of open innovation: An exploratory study. In: Proceedings of the 2012 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement. pp. 271–279 (2012)

[SP22] Vlas, R., Vlas, C.: A requirements-based analysis of success in open-source software development projects. In: AMCIS 2011 Proceedings. pp. 1–10 (2011)

[SP23] Noll, J.: Requirements acquisition in open source development: Firefox 2.0. In: Russo, B., Damiani, E., Hissam, S., Lundell, B., Succi, G. (eds.) Open Source Development, Communities and Quality. pp. 69–79. Springer US, Boston, MA (2008)

[SP24] Scacchi, W., Jensen, C., Noll, J., Elliott, M.S.: Multimodal modeling, analysis, and validation of open source software development processes. International Journal of Information Technology and Web Engineering 1(3), 49–63 (2006)

[SP25] Scacchi, W.: Understanding the requirements for developing open source software systems. IEE Proceedings - Software 149(1), 24–39 (2002)

[SP26] Tizard, J., Devine, P., Wang, H., Blincoe, K.: A software requirements ecosystem: Linking forum, issue tracker, and FAQs for requirements management. IEEE Transactions on Software Engineering pp. 1–13 (2022)

[SP27] Wang, W., Hussein, N., Gupta, A., Wang, Y.: A regression model based approach for identifying security requirements in open source software development. In: 2017 IEEE 25th International Requirements Engineering Conference Workshops (REW). pp. 443–446 (2017)

[SP28] Morales-Ramirez, I., Vergne, M., Morandini, M., Perini, A., Susi, A.: Exploiting online discussions in collaborative distributed requirements engineering. In: Castro, J., Filho, G.A.C., Liaskos, S. (eds.) Proceedings of the Eighth International i*Workshop, iStar 2015, in conjunction with the 23rd International Requirements Engineering Conference (RE 2015), Ottawa, Canada, August 24-25, 2015. CEUR Workshop Proceedings, vol. 1402, pp. 7–12. CEUR-WS.org (2015)

[SP29] Vergne, M., Susi, A.: Expert finding using markov networks in open source communities. In: Jarke, M., Mylopoulos, J., Quix, C., Rolland, C., Manolopoulos, Y., Mouratidis, H., Horkoff, J. (eds.) Advanced Information Systems Engineering. pp. 196–210. Springer International Publishing, Cham (2014)

[SP30] Alspaugh, T.A., Scacchi, W.: Ongoing software development without classical requirements. In: 2013 IEEE 21st International Requirements Engineering Conference (RE). pp. 165–174. IEEE Computer Society, Los Alamitos, CA, USA (jul 2013)

[SP31] Ernst, N.A., Murphy, G.C.: Case studies in just-in-time requirements analysis. In: 2012 Second IEEE International Workshop on Empirical Requirements Engineering (EmpiRE). pp. 25–32 (2012)

[SP32] Laurent, P., Cleland-Huang, J.: Lessons learned from open source projects for facilitating online requirements processes. In: Glinz, M., Heymans, P. (eds.) Requirements Engineering: Foundation for Software Quality. pp. 240–255. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)

[SP33] Shi, L., Chen, C., Wang, Q., Boehm, B.: Is it a new feature or simply "don't know yet"?: On automated redundant OSS feature requests identification. In: 2016 IEEE 24th International Requirements Engineering Conference (RE). pp. 377–382. IEEE Computer Society, Los Alamitos, CA, USA (sep 2016)

[SP34] Scacchi, W.: Understanding requirements for open source software. In: Design Requirements Engineering: A Ten-Year Perspective: Design Requirements Workshop, Cleveland, OH, USA, June 3-6, 2007, Revised and Invited Papers. pp. 467–494. Springer (2009)

[SP35] Salo, R., Poranen, T., Zhang, Z.: Requirements management in GitHub with a lean approach. In: SPLST. pp. 164–178 (2015)

[SP36] Gill, K.D., Raza, A., Zaidi, A.M., Kiani, M.M.: Semi-automation for ambiguity resolution in open source software requirements. In: 2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE). pp. 1–6. IEEE (2014)

[SP37] Heck, P., Zaidman, A.: Horizontal traceability for just-in-time requirements: the case for open source feature requests. Journal of Software: Evolution and Process 26(12), 1280–1296 (2014)

[SP38] Puhlfürß, T., Montgomery, L., Maalej, W.: An exploratory study of documentation strategies for product features in popular github projects. In: 2022 IEEE International Conference on Software Maintenance and Evolution (ICSME). pp. 379–383. IEEE (2022)

[SP39] Johansson, B., de Carvalho, R.A.: Management of requirements in ERP development: a comparison between proprietary and open source ERP. In: Proceedings of the 2009 ACM symposium on Applied Computing. pp. 1605–1609 (2009)

[SP40] McGrath, O.G.: Balancing act: community and local requirements in an open source development process. In: Proceedings of the 34th annual ACM SIGUCCS fall conference: expanding the boundaries. pp. 240–244 (2006)

[SP41] Bastani, B.: A requirements analysis framework for open systems requirements engineering. ACM SIGSOFT Software Engineering Notes 32(2), 1–19 (2007)

[SP42] Noll, J., Liu, W.M.: Requirements elicitation in open source software development: a case study. In: Proceedings of the 3rd International Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development. pp. 35–40 (2010)

[SP43] Vlas, R.E., Robinson, W.N.: Two rule-based natural language strategies for requirements discovery and classification in open source software development projects. Journal of management information systems 28(4), 11–38 (2012)