Challenges in Requirements Engineering and Its Solutions: A Systematic Review

Otávio da Cruz Mello and Lisandra Manzoni Fontoura
Programa de Pós-Graduação em Ciência da Computação, Federal University of Santa Maria (UFSM), Brazil

Keywords: Requirements Engineering (RE), Software Engineering (SE), Challenges, Solutions, Systematic Literature Review (SLR).

Abstract: Software development projects are susceptible to many adversities throughout their life cycle that can be originated, among several reasons, of a low-quality specification and management of requirements. To ensure the Requirements Engineering activities are conducted correctly, researchers study and apply various techniques to predict and avoid the negative impacts that may occur in projects. The main goal of this research is to identify which techniques have been used to solve problems related to requirements management in software projects. We retrieved and reviewed studies published across various scientific databases to answer research questions that were previously defined. From this work, it was possible to obtain a better understanding of the most common problems in the Requirements Engineering field, as well as some techniques that currently exist to solve them. We also identified that Artificial Intelligence has been widely explored to improve the activities of the field.

1 INTRODUCTION

Historically, it is usual for projects focused on software development to face several problems throughout their life cycle. Even after years of advances in project management practices and new methodologies, about 31% of projects tend to be discontinued, and 52% extrapolate financial resources, have deliveries after the agreed deadlines, and/or do not satisfy the needs that were promised to the customer, according to the 2019 CHAOS report (Sommerville, 2011) (Standish Group International, 2019).

Most of these problems can be originated from the way requirements are managed. Some of the essential criteria for the success of a project are the correct definition and specification of requirements in its initial stages, as well as good communication between stakeholders to ensure that everyone has a clear understanding and is satisfied with the requirements that have been specified (Iriarte and Bayona, 2020).

However, each project has different characteristics, and other types of unpredictable obstacles can generate negative impacts. To minimize possible risks, it is necessary to understand the requirements engineering process and activities and apply appropriate techniques.

Thereby, the present study aims to list the main challenges existing today in the Requirements Engineering field, from a study in literature, and explore current approaches to solve them. From this analysis, we seek to identify solutions that have not yet been explored by researchers for possible future projects.

We organized the paper as follows. In Section 2, we discuss some related research found in the literature. In Section 3, we describe the methodology and procedures for carrying out this study. We analyze and discuss the initial results obtained in Section 4. Finally, we present the conclusion and future works in Section 5.

2 RELATED WORK

Numerous researchers have described systematic reviews in the Requirements Engineering literature, mapping some of the main problems found in studies in the field.

Alam et al. (2017) performed a systematic review based on the guidelines suggested by Kitchenham et al. (2007), which consists of six basic activities: planning, identification of keywords, inclusion and exclusion, the definition of research questions, analysis and review, and description of results. The researchers analyzed studies related to Requirements Engineering in

Mello, O. and Fontoura, L.
Challenges in Requirements Engineering and Its Solutions: A Systematic Review.
DOI: 10.5220/0011079000003179
In *Proceedings of the 24th International Conference on Enterprise Information Systems (ICEIS 2022) - Volume 2*, pages 70-77
ISBN: 978-989-758-569-2; ISSN: 2184-4992

agile methodologies, seeking to answer what were the practices adopted in agile projects and what were the challenges and limitations in this same context.

Schön et al. (2017) followed the same guidelines as the previous study for their literature review. Studies related to Requirements Engineering that also addressed user and stakeholder involvement in agile methodologies were selected. The researchers investigated what approaches exist to continuously involve the stakeholder during the requirements engineering process, what methodologies are used to present the user perspective to the stakeholder, and how requirements management occurs.

Finally, Elghariani and Kama (2016), like the previous researchers, also followed the guidelines of Kitcheham et al. (2007) to carry out a systematic review of the literature related to practices and problems found in the Requirements Engineering field. Eighty studies were analyzed and twenty were selected. The researchers identified sixteen fundamental practices and six common problems of the field.

This project focuses on carrying out a systematic review of the literature to identify the main challenges of the Requirements Engineering field and also identifying existing solutions for such problems. In addition, it is also expected to determine which challenges still do not have an adequate solution, so future new research is developed to further improve the requirements specification and management activities and, this way, avoid negative impacts on a project.

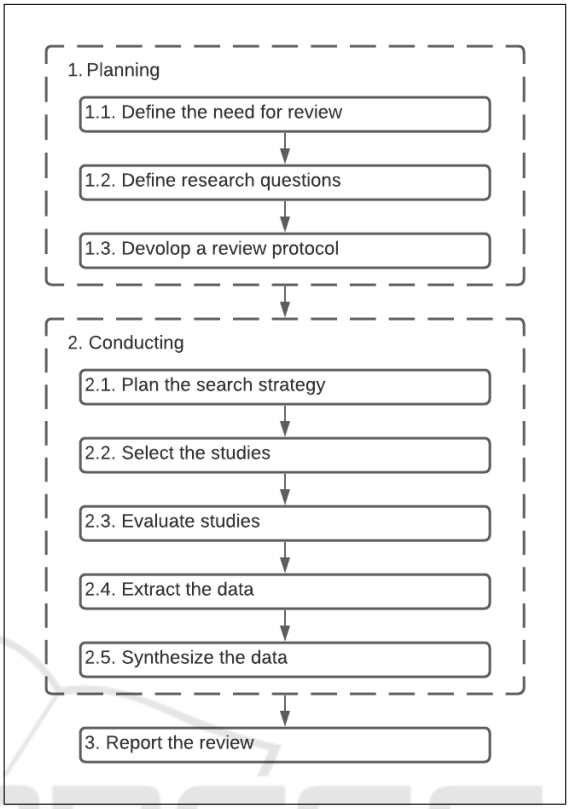

Figure 1: Systematic Literature Review Process proposed by Kitchenham et al.

3 PROCEDURES FOR THE REVIEW

To carry out the systematic literature review, we followed the guidelines and the process proposed by (Kitchenham and Charters, 2007). This process consists of three main phases, which are illustrated in Figure 1. Each step has recommended procedures and activities that must be followed so that the research can produce adequate and meaningful results.

3.1 Planning

The initial planning phase involves the preparation activities for carrying out the review. In planning, we define the goal we want to achieve with the research and the steps to reach such a goal. The steps are:

Define the Need for the Review. The need for this research originated from the problems usually faced by professionals in Requirements Engineering activities. In most cases, software development projects end up being severely affected by these problems, having impacts of great magnitude on previously estimated deadlines and costs. Thus, this study is focused on understanding what are the most frequent difficulties in this area and also what solutions currently exist to prevent, or at least reduce, these difficulties.

Define Research Questions. Research questions help the researcher focus his study, answering specific questions defined by him to achieve his goal. For the present work, we defined three research questions to be answered.

- **Q_1:** What are the most frequently faced challenges in Requirements Engineering currently?
- **Q_2:** What approaches are proposed to prevent or reduce such difficulties?
- **Q_3:** What opportunities exist for future research according to the current scenario?

To answer research question Q_1, we analyzed works published between 2015 and 2021 reporting problems experienced in Requirements Engineering during the development of software projects. After mapping these problems, we identified in the literature techniques and approaches that currently exist to

solve them to answer the research question Q_2 Finally, we answered question Q_3 based on the two previous research questions. We analyze which areas are open, or can still be explored, to improve activities of the field and reduce risks.

3.2 Conducting

From the definition of the review protocol, which establishes methods and procedures that should be taken when selecting and analyzing studies, it is possible to start the execution of the activities that were planned in the previous phase. Conducting the systematic literature review is composed of the following steps:

Define Search Terms. The studies were retrieved from the IEEE Xplore, ACM Digital Library, and ScienceDirect scientific databases. To filter the most relevant publications to the present study, we defined search terms that take into account what was needed to answer the research questions. The result was the following search string:

("requirements engineering" OR "requirements management" OR "requirements elicitation" OR "requirements analysis" OR "requirements specification" OR "requirements validation" OR "software requirements") AND ("challenges" OR "issues" OR" obstacles" OR "difficulties" OR "burden" OR "problems" OR "complications" OR "hardship")

Therefore, the string limits the search to publications that address specific problems related to the main activities of Requirements Engineering, as well as the field as a whole.

Define Inclusion and Exclusion Criteria. Inclusion and exclusion criteria filter publications that will be relevant during the review process. This search used the following criteria:

Inclusion Criteria.

- Studies published between 2015 and 2021.
- The text must be available in full.
- The studies are peer-reviewed publications.
- The studies are relevant to the defined search terms.
- The studies describe the challenges related to Requirements Engineering experienced during the development of a software project.

Exclusion Criteria.

- Studies that are not focused on or do not discuss requirements engineering topics.
- Studies that describe neither challenges nor solutions experienced in Requirements Engineering.
- Publications like *Keynotes*, *White Papers*, or works consisting only of the *abstract*.
- Publications that are not in English.
- Studies that do not meet any inclusion criteria that have been defined.
- Duplicate studies.

Study Selection. The first selection was performed by reading the title and *abstract* of the publications to understand if they were aligned with the proposed theme. The first selection resulted in 152 results.

After the initial selection, a more detailed analysis was carried out in two stages, in which the publications were read and the inclusion and exclusion criteria were applied. 36 of the 152 publications selected initially were considered acceptable for review.

The number of publications retrieved from the execution of the search string in the three defined bases and the number of filtered studies in each stage can be seen in Table 1.

Table 1: Selected studies.

Base	1st Sel.	2nd Sel.	Final Sel.
IEEE Xplore	32	14	10
ACM Library	49	24	11
ScienceDirect	71	35	15
Total	152	73	36

4 INITIAL RESULTS AND DISCUSSION

This section focuses on the exposure of the initial results obtained from the systematic review of the literature we performed. To answer the defined research questions, we identified in the selected studies the main challenges faced in Requirements Engineering activities and then classified each problem into related categories. In the end, we calculated the frequency of each challenge based on its occurrences in studies.

The mapping of the most frequent challenges is available in Table 2. It helps to answer the first research question $\mathbf{Q_1}$. It is possible to see that the lack of communication between the stakeholders is the most cited challenge in Requirements Engineering among the selected publications, occurring in about 56% of the cases.

Also with considerably high percentages are ambiguous, incomplete, inconsistent, or incorrect requirements with 33%, 31%, 22%, and 19% of occurrence, respectively. Another frequent problem, the

Table 2: The main Requirements Engineering challenges according to the analysis of selected publications.

Challenge	Frequency (Total: 36)	%
Lack of communication and/or between stakeholders	20	56
Ambiguous requirements	12	33
Incomplete requirements	11	31
Documentation	10	28
Requirements traceability	9	25
Inconsistent requirements	8	22
Insufficient knowledge about the domain	8	22
Requirements priorization	8	22
Incorrect requirements	7	19
Volatile requirements	7	19
Technical difficulties	5	14

maintenance of documentation, appears next, cited in 28% of the studies as a challenge.

In addition to those already mentioned, other adversities tend to generate complications to the project, such as insufficient knowledge about the application domain (22%), prioritization of requirements (22%), and changing requirements (19%). A less frequent challenge is general technical difficulties, which generally involve a lack of techniques and tools to manage requirements, with only 14% of occurrences.

For each problem identified, we searched the literature for some of the most current and effective techniques that exist to try to solve them and thus answer the research question $\mathbf{Q_2}$.

C_1: Lack of Communication and/or Involvment between the Stakeholders

Although communication and engagement depend on stakeholder availability, there are some traditional methodologies to improve team communication as the MUST method, Joint Application Design (JAD), User-Led Requirements Construction (ULRC), and Soft Systems Methodology (SSM). These are focused on defining best practices for requirements elicitation (Coughlan and Macredie, 2002).

Currently, many publications propose innovative ways to make this communication happen effectively in projects of different characteristics. In global software development projects, for example, the lack of communication between the parties involved is a common problem, as the team and the client are distributed in many places around the planet. Thus, Nadeem and Lee (2019) proposed a framework based on Case-Based Reasoning (CBR) for knowledge storage and elicitation techniques to improve the dialogue between stakeholders and thus define best practices for requirements gathering in the context of global software development.

In a similar study, Shahzad et al. (2021) discuss the necessary changes for requirements elicitation tasks in the context of the global COVID-19 pandemic and propose the use of blockchain-based technologies to improve various challenges related to the Requirements Engineering field, including communication between stakeholders, conflicts in decision-making, negotiations, among others.

C_2: Ambiguous Requirements

The detection of ambiguous requirements is currently highly explored by researchers, who mainly use intelligent algorithms to correct already specified documents. Among the specific fields of Artificial Intelligence, Machine Learning is perhaps one of the most frequent in works related to requirements ambiguities reduction.

Osman and Zaharin (2018) proposed a hybrid Text Mining and Machine Learning approach to detect and classify ambiguities in a dataset extracted from Malaysian specification documents. From the extraction of information in the mining stage, the system progressively learns to detect ambiguous requirements.

Also focused on Machine Learning, Sharma et al. (2016) proposed an approach for detecting harmful pronomial anaphor ambiguities in requirements specification documents. The researchers used classification algorithms to classify harmful and non-harmful ambiguities. The chosen classifier was able to correctly detect 95% of harmful ambiguous requirements.

C_3: Incomplete Requirements

As well as ambiguous requirements, incomplete requirements are another common challenge in requirements engineering and can be caused by the precarious definition of requirements in the specification stage. It is often difficult to identify requirements incompleteness before functionality is developed, especially when documents are not periodically reviewed. Thus, it is interesting to use tools that help the professional in this task.

DeVries and Cheng (2017) proposed the Ares-C approach for detecting incomplete requirements, using symbolic analysis and evolutionary computation to analyze hierarchical requirements models. This approach was applied to a real system, and the researchers were able to notice that Ares-EC was able to automatically detect incomplete requirements and generate completeness counter-examples.

Kalinowski et al. (2016) conducted a questionnaire in 88 small, medium, and large-sized Brazilian and Austrian organizations to define the most com-

mon causes for incomplete or hidden requirements in a project. Focusing on these causes, the researchers discussed mitigation practices and actions based on the responses obtained from the respondents.

C_4: Documentation
Having documentation that is not accessible to all parties involved, that is not managed properly, or that simply does not exist are major risk factors for a project. It is necessary to have good practices defined to document and manage the documents to ensure the software product quality.

Salvador and Santos (2016) presented the DoMaR application in their study, which is focused on preventing problems related to requirements management and documentation by mapping these problems through questionnaires carried out with experts. Based on the experts feedback, their application showed potential to solve the challenges frequently mentioned in the questionnaires answered.

Behutiye et al. (2017) interviewed professionals from four different companies and presented their findings regarding the documentation practices adopted by organizations. Based on their analysis, the researchers proposed guidelines for documenting requirements in agile software development, in which documents are generally not given priority.

C_5: Requirements Traceability
Traceability is the ability to follow a requirement from its origin and specification through its development and implementation. Ensuring traceability throughout the software development cycle is a complex task, but it helps maintain and control project requirements.

To propose a solution to the traceability problem, Haidrar et al. (2018) developed a language for requirements specification named ReqDL. The language has specific operators that reveal the explicit and implicit links between requirements and artifacts. From the use of ReqDL expressions, the traceability task becomes less complex and more understandable.

Garcia and Paiva (2016) presented a tool to help the user keep the traceability information of the requirements updated, without errors. By using the developed tool, the researchers demonstrated that creating a link between requirements and implementation artifacts was more effective in maintenance than traceability matrices, which are standard documents to show the relationship between artifacts in Requirements Engineering.

C_6: Inconsistent Requirements
Consistency between requirements is yet another crucial aspect to maintain the quality of software product development. The inconsistency goes unnoticed most of the time and can cause damages to the project, such as rework and wasted effort, generating an increase in time and resources needed for a project to be completed.

To address requirements inconsistencies in the context of global software development, Gull et al. (2021) introduced the BOMO framework based on blockchain technology and Model-Driven Software Engineering (MDSE) methodology. This union makes it possible to manage inconsistent requirements effectively and easily. The results of the case study applying the framework proved to be positive, helping the global software company to deal with the inconsistency.

Mezghani et al. (2018) used the unsupervised Machine Learning algorithm, k-means, to identify redundancy and inconsistency in requirements. The k-means algorithm groups the data around centroids, according to a defined value of groups named ”k”. The algorithm was validated using a real industrial base containing inconsistent data and using a number defined by a Requirements Engineer who assisted in the research as the value of ”k”. The experiments generated positive initial results in detecting inconsistent and redundant requirements.

C_7: Insufficient Knowledge about the Application Domain
An RE specialist who does not know the application domain tends to fail to perform requirements specification and management activities correctly. Understanding the domain of an application to be developed is a task that demands time and dedication from the professional involved in Requirement Engineering activities and, for this reason, applying means of obtaining this knowledge in a simple and fast way in the project can bring beneficial results.

Li et al. (2020) proposed an automated way of extracting domain knowledge from requirements documents to help professionals. The tool represents the documents in natural language as a vector using the Doc2Vec algorithm and then applies clustering algorithms to create the initial cluster feature tree. Extracted results containing the most important words and phrases are returned to the user for analysis.

C_8: Requirements Priorization
Prioritizing requirements is a task that plays a big role during the implementation phase. Prioritizing features is a highly complex process and takes into account aspects such as the team’s ability to develop at the moment and the customer’s urgency for the requirement.

There are currently several ways to perform re-

quirements prioritization, some known to produce better results. However, there is no consensus on which is the best technique to prioritize requirements, since it all depends on the characteristics of the project, the team, and the product. Studies proposing new ways to prioritize focusing on usability, scalability, and quality assurance are often published.

In this perspective, Mkpojiogu and Hashim (2017) proposed a quality-based approach to requirements prioritization, using the Kano model. Requirements are prioritized according to attributes that take into account the stakeholders' point of view on quality. The result showed that the model generated consistent results and that it can serve as a good option for requirements prioritization.

Yaseen et al. (2020) suggested a Spanning Tree-based approach for prioritizing requirements from the developer's perspective. The approach was validated using a set of requirements from the ERP ODOO software, which were assigned to four developers to have dependencies. The researchers then performed the time estimates for the prioritized and non-prioritized requirements to assess the impact on the total project estimate time. A significant difference was noticed between the time estimates of prioritized and non-prioritized requirements, which demonstrated the importance of prioritizing requirements.

C_9: Incorrect Requirements

Incorrect requirements are generally wrong definitions of functionalities required by the customer, or definitions that conflict with previously specification documents about the product. It can happen due to an error of the specialist when preparing the requirements specification document or due to a misunderstanding about functionalities. An incorrect requirement can lead to a product that does not meet the customer's needs and requires reworking.

To improve the quality of the requirements analysis step, Nguyen et al. (2014) developed the GUITAR tool to detect incorrect, incomplete, and inconsistent artifacts. The GUITAR approach is based on domain ontologies and semantics for analysis of requirements. The core of this methodology is the representation of activities, which are composed of both an action and an object ("create review", for example). These activities are related to one another, thus it is easier to identify incompatibilities between artifacts or missing artifacts from these relationships.

C_{10}: Volatile Requirements

Changes are inevitable within any project. It is impossible to define all that is expected from software at the beginning of development, so the customer must make frequent change requests to the team (Sommerville and Kotonya, 1998). These requests may involve changing an existing requirement in the project or incorporating new requirements that were not previously planned.

Thus, to minimize negative impacts throughout its life cycle, the project must have a well-defined Change Management Process, which does not present conflicts to its practices.

Ali et al. (2018) proposed a framework for managing requirements changes based on the Case-Based Reasoning technique in the context of global software development. With the application of the CBR-based framework in the cloud, the authors noticed that the communication and coordination of the global team during change management, which was previously challenging, became more effective. All services required became available for the users at all times on a single platform without time and space restrictions, when previously the team used tools on different platforms with different logins, not taking cultural differences into account.

Naz et al. (2013) defined and described a model that integrates requirements change management with the Case-Based Reasoning technique. For evaluation, the researchers presented their model to experts in the field and asked about the experience gained once the model was implemented, thus comparing the performance differences before and after its use. The results show that the framework helped to reduce the cost and time impacts of a change, as well as to resolve requirements conflicts and increase customer satisfaction.

Regarding the C_{11} challenge, which refers to technical difficulties, there are no specific solutions to be used. Such challenge refers to general problems related to the organization's practices, such as the lack of clearly defined tools and processes for managing requirements.

Looking at the current scenario, as defined in the research question **Q_3**, it is clear that researchers are exploring innovative ways to solve classic requirements engineering problems. It is possible to notice that the scientific community is looking for solutions in other areas, such as Artificial Intelligence, which was present in most of the publications studied, as seen in Figure 2. Many studies also suggested best practices for Requirements Engineering activities (28%), while some proposed approaches based on Blockchain technology (11%). Other types of techniques were not as frequent.

According to (Harman, 2012), the relationship between AI and Software Engineering areas tends to generate beneficial results. Software Engineering is a field highly focused on knowledge, but uncertainty is

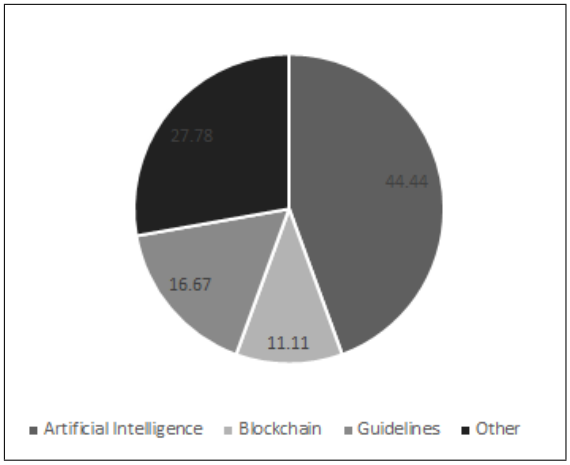

Figure 2: Most frequent approaches in the studied solutions.

its main obstacle. From the techniques and algorithms provided by Artificial Intelligence, such uncertainty can be reduced.

Analyzing the number of studies containing the keywords "Artificial Intelligence" and "Software Engineering" published between 2010 and 2021 in the IEEE Xplore, ACM Digital Library, and ScienceDirect databases, one can also see a growing trend in the intersection of these two areas. As illustrated in Figure 3, in 2010 about 900 publications addressed both topics, while in 2021 this number exceeded the 2500 mark. Analyzing the graph, the number of publications mentioning AI and Software Engineering today is almost three times greater than it was in 2010.

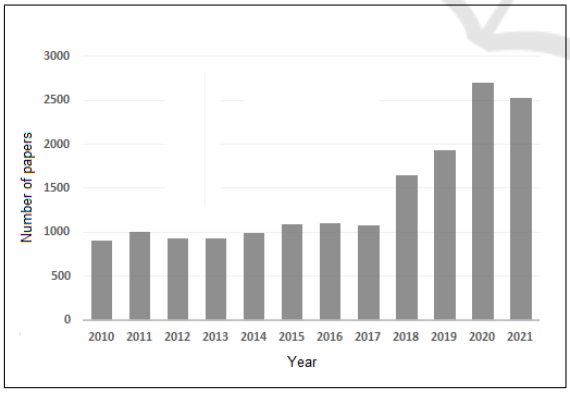

Figure 3: Number of publications addressing Software Engineering and Artificial Intelligence, per year.

Based on this information, it is possible to note that Artificial Intelligence is a branch of science that has much to offer to the Software Engineering field. As it is a vast and constantly growing area, there are still many open possibilities of also applying intelligent techniques in Requirements Engineering.

5 CONCLUSION

Requirements Engineering is a subarea of Software Engineering that encompasses activities such as analysis, elicitation, documentation, and requirements management and is crucial for the success of a project. Projects that do not have good techniques and practices defined to perform such activities tend to face problems throughout their life cycle and consequently extrapolate initially estimated deadlines and resources.

We carried out a systematic literature review to define the most frequent problems faced in Requirements Engineering today. Among the challenges identified, the ones that had the highest occurrence were: lack of communication between stakeholders, ambiguous, incomplete, and inconsistent requirements, problems related to documentation, requirements traceability, insufficient knowledge about the domain, and requirements prioritization.

In addition, for each challenge encountered, we studied the current scenario concerning published solution proposals. We understand that, currently, Requirements Engineering still presents numerous challenges, and researchers are continually proposing innovative solutions to make activities in this field less problematic, even intersecting RE concepts with other areas of science.

In future work, we plan to use the knowledge acquired by the elaboration of this work and apply them in the development of a solution proposal for the most common problems of Requirements Engineering.

THREATS TO VALIDITY

The fact that only a small set of studies were selected for this systematic literature review poses a potential threat to its validity. Many studies also don't go in depth about problems faced, which could possibly lead to the misinterpretation of such problems.

ACKNOWLEDGEMENTS

We thank the Brazilian Army Strategic Program ASTROS for the financial support through the SIS-ASTROS GMF (898347/2020) projects.

REFERENCES

Alam, S., Nazir, S., and Asim, S. (2017). Impact and challenges of requirement engineering in agile methodolo-

gies: A systematic review. *International Journal of Advanced Computer Science and Applications*, 8.

Ali, S., Iqbal, N., and Hafeez, Y. (2018). Towards requirement change management for global software development using case base reasoning. *Mehran University Research Journal of Engineering and Technology*, 37.

Behutiye, W., Karhapää, P., Costal, D., Oivo, M., and Franch, X. (2017). Non-functional requirements documentation in agile software development: Challenges and solution proposal. pages 515–522.

Coughlan, J. and Macredie, R. (2002). Effective communication in requirements elicitation: A comparison of methodologies. *Requirements Engineering*, 7:47–60.

DeVries, B. and Cheng, B. (2017). Automatic detection of incomplete requirements using symbolic analysis and evolutionary computation. In *SSBSE*.

Elghariani, K. and Kama, N. (2016). Review on agile requirements engineering challenges. pages 507–512.

Garcia, J. E. and Paiva, A. C. (2016). A requirements-to-implementation mapping tool for requirements traceability. *J. Softw.*, 11:193–200.

Gull, N., Rashid, M., Azam, F., Rasheed, Y., and Waseem Anwar, M. (2021). A block-chain oriented model driven framework for handling inconsistent requirements in global software development. In *2021 10th International Conference on Software and Computer Applications*, ICSCA 2021, page 105–111, New York, NY, USA. Association for Computing Machinery.

Haidrar, S., Anwar, A., Bruel, J., and Roudiès, O. (2018). A domain-specific language to manage requirements traceability. *J. Softw.*, 13:460–480.

Harman, M. (2012). The role of artificial intelligence in software engineering. In *RAISE '12: Proceedings of the First International Workshop on Realizing AI Synergies in Software Engineering*, pages 1–6, Zurich, Switzerland. IEEE Press.

Iriarte, C. and Bayona, S. (2020). It projects success factors: a literature review. *International Journal of Information Systems and Project Management*, 8:49–78.

Kalinowski, M., Felderer, M., Conte, T., Spínola, R., Prikladnicki, R., Winkler, D., Méndez Fernández, D., and Wagner, S. (2016). Preventing incomplete/hidden requirements: Reflections on survey data from austria and brazil.

Kitchenham, B. and Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. 2.

Li, Y., Schulze, S., Scherrebeck, H. H., and Fogdal, T. S. (2020). Automated extraction of domain knowledge in practice: The case of feature extraction from requirements at danfoss. In *Proceedings of the 24th ACM Conference on Systems and Software Product Line: Volume A - Volume A*, SPLC '20, New York, NY, USA. Association for Computing Machinery.

Mezghani, M., Kang, J., and Sedes, F. (2018). Industrial requirements classification for redundancy and inconsistency detection in semios. pages 297–303.

Mkpojiogu, E. O. C. and Hashim, N. (2017). Quality based prioritization: An approach for prioritizing software requirements. *Journal of Telecommunication, Electronic and Computer Engineering*, 9:17–21.

Nadeem, M. and Lee, S. U.-J. (2019). Requirement elicitation framework for global software development. *Indian journal of science and technology*, 12:1–6.

Naz, H., Motla, Y., Asghar, S., Abbas, M., and Khatoon, A. (2013). Effective usage of ai technique for requirement change management practices. pages 121–125.

Nguyen, H., Grundy, J., and Almosy, M. (2014). Guitar: An ontology-based automated requirements analysis tool.

Osman, M. H. and Zaharin, M. F. (2018). Ambiguous software requirement specification detection: An automated approach. In *Proceedings of the 5th International Workshop on Requirements Engineering and Testing*, RET '18, page 33–40, New York, NY, USA. Association for Computing Machinery.

Salvador, M. E. R. F. L. and dos Santos, L. B. R. (2016). Domar: An approach to prevent problems related to requirements documentation and management.

Schön, E.-M., Thomaschewski, J., and Cuaresma, M. J. E. (2017). Agile requirements engineering: A systematic literature review. *Comput. Stand. Interfaces*, 49:79–91.

Shahzad, B., Javed, I., Shaikh, A., Sulaiman, A., Abro, A., and Ali Memon, M. (2021). Reliable requirements engineering practices for covid-19 using blockchain. *Sustainability*, 13(12).

Sharma, R., Sharma, N., and Biswas, K. K. (2016). Machine learning for detecting pronominal anaphora ambiguity in nl requirements. *2016 4th Intl Conf on Applied Computing and Information Technology/3rd Intl Conf on Computational Science/Intelligence and Applied Informatics/1st Intl Conf on Big Data, Cloud Computing, Data Science & Engineering (ACIT-CSII-BCD)*, pages 177–182.

Sommerville, I. (2011). *Software Engineering*. Pearson Prentice Hall, São Paulo, SP, Brasil, 9th edition.

Sommerville, I. and Kotonya, G. (1998). *Requirements Engineering - Processes and Techniques*. Wiley, Hoboken, NJ, United States, 1st edition.

Standish Group International (2019). The standish group report chaos.

Yaseen, M., Mustapha, A., and Ibrahim, N. (2020). Prioritization of software functional requirements: Spanning tree based approach. *International Journal of Advanced Computer Science and Applications*, 10.